NASA Contractor Report 182016
ICASE Report No. 90-25

ICASE

HIGH ORDER FILTERING METHODS FOR APPROXIMATING HYPERBOLIC SYSTEMS OF CONSERVATION LAWS

F. Lafon
S. Osher

Contract No. NAS1-18605
March 1990

Institute for Computer Applications in Science and Engineering
NASA Langley Research Center
Hampton, Virginia 23665-5225

Operated by the Universities Space Research Association

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665-5225

HIGH ORDER FILTERING METHODS FOR APPROXIMATING HYPERBOLIC SYSTEMS OF CONSERVATION LAWS

F. Lafon[1] & S. Osher[2]
Department of Mathematics
University of California
Los Angeles, CA 90024-1555

ABSTRACT

In the computation of discontinuous solutions of hyperbolic systems of conservation laws, the recently developed ENO (Essentially Non-Oscillatory) schemes appear to be very useful. However, they are computationally costly compared to simple central difference methods. In this paper we develop a filtering method which uses simple central differencing of arbitrarily high order accuracy, except when a novel local test indicates the development of spurious oscillations. At these points, generally few in number, we use the full ENO apparatus, maintaining the high order of accuracy, but removing spurious oscillations. Numerical results indicate the success of the method. We obtain high order of accuracy in regions of smooth flow without spurious oscillations for a wide range of problems and a significant speed up of generally a factor of almost three over the full ENO method.

[1]Research supported by grant N00014-86-K-0691 and NSF grant DMS 88-11863.

[2]The research for the second author was partially supported by the National Aeronautics and Space Administration under NASA Contract No. NAS1-18605 while he was in residence at the Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, Virginia, 23665. Research was also supported by grant N00014-86-K-0691, DARPA grant in the ACMP program, NFS grant DMS 88-11863, and NASA Langley NAGI-270.

1 Introduction

Recently, a new class of high order schemes related to numerical calculations of linear and nonlinear systems of conservation laws has been developed. This new class of methods uses an explicit TVD Runge-Kutta Multistage time discretization together with high order Essentially Non Oscillatory (ENO) spatial discretization methods. Details of these methods can be found in [6, 7] and in references quoted therein. These methods are applied to solve numerically hyperbolic systems of conservation laws

$$\mathbf{u}_t + (\mathbf{f}(\mathbf{u}))_x = 0, \tag{1}$$

$$\mathbf{u}(x,0) = \mathbf{u}_0(x), \tag{2}$$

to be solved for $t > 0$ and x in some interval Ω with appropriate boundary conditions. An example of such a system of conservation laws is given by the Euler equations of compressible gas dynamics for which

$$\mathbf{f}(\mathbf{u}) = v\mathbf{u} + (0, p, vp)^T \tag{3}$$

and $\mathbf{u} = (\rho, q, p)$, ρ is density, q is momentum, v is velocity, and p is the pressure. A similar example for two dimensional flows will be considered. The two dimensional version of the equation (1) now has different fluxes for each space dimension. We have:

$$\mathbf{u}_t + (\mathbf{f}(\mathbf{u}))_x + (\mathbf{g}(\mathbf{u})_y = 0, \tag{4}$$

$$\mathbf{u}(x,y,0) = \mathbf{u}_0(x,y), \tag{5}$$

to be solved for $t > 0$, $(x,y) \in \Omega$, some compact set, with appropriate boundary conditions. The fluxes are $\mathbf{f}(\mathbf{u}) = v_x\mathbf{u} + (0, p, 0, v_x p)^T$ and $\mathbf{g}(\mathbf{u}) = v_y\mathbf{u} + (0, 0, p, v_y q)^T$, respectively, where $\mathbf{u} = (\rho, q_x, q_y, e)^T$. In numerical experiments, we approximate the solution of equations (1) or (4) by using point values. That is, $\mathbf{u}(x_j, t^n)$ is approximated by $\mathbf{u}_j^n$, given a regular triangulation of the domain Ω. In this paper, only a line by line discretization will be considered, restricting the shape of domain Ω to regions which can be mapped onto squares or rectangles.

The TVD time discretization performed is the one introduced in [6, 7]. The method is explicit and relatively easy to program. Such algorithms can be briefly described as follows:

$$\mathbf{u}^{n+i/m} = \sum_{k=0}^{i-1} [\alpha_{i,k}\, \mathbf{u}^{n+k/m} + \beta_{i,k}\Delta t L(\mathbf{u}^{n+k/m})], \tag{6}$$

where m is the number of stages to move the solution from time t to $t + \Delta t$. Generally, second to sixth order methods are investigated. The coefficients $\alpha_{i,k}$ and $\beta_{i,k}$ are calculated

so as to improve the CFL number in order to minimize the number of time iterations. In the class of method defined by (6), the CFL coefficient λ must satisfy:

$$\lambda \leq \min_{i,k} \frac{\alpha_{i,k}}{|\beta_{i,k}|}.\lambda_0,$$

where λ_0 is the maximum allowable value for the forward Euler method (see [6]). In particular, it is possible to obtain a third order accurate TVD time discretization method with a CFL number of 1. This is only slightly reduced (to $\frac{2}{3}$) for the fourth order method. In addition, it is possible to derive a class of time discretizations that require the evaluation of $L(\mathbf{u}^{n+(i-1)/k})$ only, so that $\beta_0, ..., \beta_{i-2} = 0$. This process reduces the storage requirement significantly. However, such procedure is possible only for methods of up to third order accuracy. For higher order methods, several evaluations of the operator L are needed to enforce the TVD property; see [6, 7] for more details.

For the space discretization, we use high order ENO methods to approximate $L(\mathbf{U})$ using conservation form.

$$-L(\mathbf{u}_j) = \frac{\mathbf{f}(\mathbf{u}_{j-r+1/2}, \cdots, \mathbf{u}_{j+s+1/2}) - \mathbf{f}(\mathbf{u}_{j-r-1/2}, \cdots, \mathbf{u}_{j+s-1/2})}{\Delta x}. \tag{7}$$

For multidimensional operator, $-L = \mathbf{f}_x + \mathbf{g}_y$, $-\mathbf{f}_x$ is performed as in (7), and $-\mathbf{g}_y$ is approximated analogously. For systems of equations, a field by field decomposition is used. We calculate at each point the eigen decomposition of different fluxes, evaluating the r^{th} order accurate interpolating polynomial that approximates the fluxes in each field, and then recover each vector field in the physical space by inverse decomposition. In many cases (Euler equations for example), the decomposition in each field uses the left eigenvectors $\mathbf{L}_k$, so that $\mathbf{f}_{k,j+1/2} = \mathbf{L}_k.\mathbf{A}(\mathbf{u}_{j,j+1})$, where $\mathbf{A}$ is the Roe matrix for $\nabla \mathbf{f}(\mathbf{u})$ (see [10] for more details). The ENO algorithm is based on a Newton interpolating polynomial using an adaptative stencil. That is, instead of considering a polynomial interpolant using a fixed centered or a fixed upwind stencil, we derive an interpolating polynomial minimizing the successive undivided differences. This process limits oscillations, thus the name of the method. In the case of a shock discontinuity, this method works quite well, leading to sharp transitions over a few points. However, smearing of linear (e.g contact) discontinuities may occur. In this case, a particular treatment using subcell resolution or artificial compression is available to sharply resolve large transitions. The interested reader can find more details in [4, 7, 12], we do not use this improvement here.

The reader has probably already realized that such methods require a considerable amount of computational time when multidimensional systems are investigated. It would be interesting to use high order methods derived from simple spatial discretization most of the time, and use ENO methods only when spurious oscillations appear. Of course, it is well

known that such oscillations will generally occur for nonlinear equations even with smooth initial conditions.

In order to deal with problems in which singularities occur but still use simple interpolation techniques, hence simple finite difference methods, we introduce a postprocessing step that detects existing singularities and spurious oscillations and corrects them if necessary. Such a method based on a postprocessing step has been introduced in [1, 11]. It relies on a simple type of correction by pushing points up or down up to an acceptable level so that the global solution satisfies both total variation diminishing (TVD) and conservation properties. By modifying this type of filter, the third author in [1] was able to prove convergence, in [11], to the physical solution for one dimensional nonlinear conservation laws. His proof relies on compensated compactness arguments using Young measures. Some references can be found in [11] and in other papers quoted therein. More importantly, an extremely simple but useful TVD algorithm which works very well in practice was developed in [1, 11].

In this paper, we define a new class of filters enforcing uniformly high order of accuracy without allowing significant spurious oscillations. Section 2 is devoted to a detailed presentation of our method . Section 3 offers several numerical examples in one and two space dimensions for both scalars and systems of conservation laws. Section 4 will conclude this paper and propose several different approaches for solving nonlinear problems in general domains using finite element grids.

2 High order uniform filtering methods

We now outline our filtering method. We first only consider spatially centered differences. This process leads to a simple scheme with low computational cost for the evaluation of the numerical fluxes. Then, from the solution which has been evaluated from this basic simple scheme, we perform another step that filters the numerical oscillations by using high order accurate ENO interpolation. We then use a high order TVD-RK time scheme. The algorithm is thus simply:

- **For** $i = 1, m$ **do**

  1. Approximate the equation (1) or (4) using the basic scheme

$$\mathbf{v}^{n+i/m} = \mathbf{D}(\mathbf{u}^n, \cdots, \mathbf{u}^{n+(i-1)/m}), \tag{8}$$

  where $\mathbf{D}$ is the numerical operator that performs the m-multistage TVD-RK algorithm as in [6] together with centered spatial differences approximating the operator L in (7) in conservation form.

2. If $\mathbf{v}^{n+i/m}$ has spurious oscillations **Then** we correct it using the filter:

$$\mathbf{u}^{n+i/m} = \mathbf{F}(\mathbf{u}^n, \cdots, \mathbf{u}^{n+(i-2)/m}, \mathbf{v}^{n+(i-1)/m}), \tag{9}$$

where $\mathbf{F}$ is the numerical operator that uses the same time discretization algorithm as the basic scheme together with a high uniform ENO type filter in evaluating numerical fluxes.

- **End For.**

We define high order $2p^{th}$ approximations as follows:

- **Second Order:** $(p = 1)$

$$\mathbf{f}_{j+1/2} = \frac{\mathbf{f}(\mathbf{u}_{j+1}) + \mathbf{f}(\mathbf{u}_j)}{2}, \tag{10}$$

- **Fourth Order:** $(p = 2)$

$$\mathbf{f}_{j+1/2} = \frac{7}{12}(\mathbf{f}(\mathbf{u}_{j+1}) + \mathbf{f}(\mathbf{u}_j)) - \frac{1}{12}(\mathbf{f}(\mathbf{u}_{j+2} + \mathbf{f}(\mathbf{u}_{j-1})), \tag{11}$$

- **Higher Order $\forall p$:** It is a simple matter, using Richardson's extrapolation to construct and obtain arbitrary high order accurate centered difference methods. This has been done in many places, e.g [5]. We will have:

$$\begin{aligned} \mathbf{f}_{j+1/2} &= \sum_{i=-p+1}^{p} \beta_i \mathbf{f}(\mathbf{u}_{j+i}), \\ \text{where } \beta_{-i+1} &= \beta_i \text{ for } i = -p+1, ..., p-1, \\ \text{and } \sum_{i=-p+1}^{p} \beta_i &= 1. \end{aligned}$$

For the time discretization, we consider the TVD Runge-Kutta type idea introduced in [6, 7], that is

- **For second order method:** $(p = 1)$

$$\mathbf{u}^{n+1/2} = \mathbf{u}^n + \Delta t L(\mathbf{u}^n), \tag{12}$$

$$\mathbf{u}^{n+1} = \frac{1}{2}\mathbf{u}^{n+1/2} + \frac{1}{2}\mathbf{u}^n + \Delta t L(\mathbf{u}^{n+1/2}). \tag{13}$$

- **For Third order method:** $(p = 3/2)$

$$\mathbf{u}^{n+1/3} = \mathbf{u}^n + \Delta t L(\mathbf{u}^n), \tag{14}$$

$$\mathbf{u}^{n+2/3} = \frac{3}{4}\mathbf{u}^n + \frac{1}{4}\mathbf{u}^{n+1/3} + \frac{1}{4}\Delta t L(\mathbf{u}^{n+1/3}), \tag{15}$$

$$\mathbf{u}^{n+1} = \frac{1}{3}\mathbf{u}^n + \frac{2}{3}\mathbf{u}^{n+2/3} + \frac{2}{3}\Delta t L(\mathbf{u}^{n+2/3}). \tag{16}$$

- Higher order methods of this type are described in [6] up to sixth order.

The filter step changes the centered differences spatial approximation to the more stable ENO approximation of fluxes. As building block, we use either the Roe scheme (see [10, 7]) which admits expansion shocks at sonic points or the Local Lax-Friedrichs decomposition of the fluxes at such points, see [7]. Both building blocks first decide on the initial stencil that respects the local characteristic direction and then evaluate the polynomial interpolant using an adaptative stencil. This stencil is chosen in order to minimize derivatives of the interpolating polynomial. The algorithm for computing the numerical fluxes in the filtering step is precisely the algorithm 2.3, in [7]. Furthermore, in order to still get a globally conservative scheme, backward and forward corrections are performed. This lead to these four possible approximations of $\mathbf{u}_j^{n+i/m}$ after the filtering step:

$$\begin{aligned}
\mathbf{u}_j^{n+i/m} &= A(\mathbf{u}_j^n, ..., \mathbf{u}_j^{n+(i-1)/m}) + \frac{\Delta t}{\Delta x}(\mathbf{f}_{j+1/2}^{eno} - \mathbf{f}_{j-1/2}^{eno}), \\
\mathbf{u}_j^{n+i/m} &= A(\mathbf{u}_j^n, ..., \mathbf{u}_j^{n+(i-1)/m}) + \frac{\Delta t}{\Delta x}(\mathbf{f}_{j+1/2}^{eno} - \mathbf{f}_{j-1/2}^{c}), \\
\mathbf{u}_j^{n+i/m} &= A(\mathbf{u}_j^n, ..., \mathbf{u}_j^{n+(i-1)/m}) + \frac{\Delta t}{\Delta x}(\mathbf{f}_{j+1/2}^{c} - \mathbf{f}_{j-1/2}^{eno}), \\
\mathbf{u}_j^{n+i/m} &= A(\mathbf{u}_j^n, ..., \mathbf{u}_j^{n+(i-1)/m}) + \frac{\Delta t}{\Delta x}(\mathbf{f}_{j+1/2}^{c} - \mathbf{f}_{j-1/2}^{c}),
\end{aligned}$$

for $i = 1, ..., m$. The operator A represents some linear combination of $\mathbf{u}^{n+k/m}$, for $k = 0, ..., i-1$, given for example by the equations (12),(13), or (14),(15),(16), and $\mathbf{f}^c$, $\mathbf{f}^{eno}$ are the computed fluxes using centered differences or ENO interpolants, respectively. If the evaluation of $\mathbf{f}_{j-1/2}^{eno}$ is needed, then a correction is performed on $\mathbf{u}_{j-1}^{n+i/m}$, where j is a multiple index in the case of several dimensions. Hopefully, for regular grids, backward corrections will only occur at the first column and first row of the computational domain Ω. Hence, at interior grid points, only forward corrections of the fluxes are performed. For example, in two dimensions, an initialization step in the filtering step is done for the first column and first row, say $i = i_0$, and $j = j_0$, leading to corrections of the fluxes $\mathbf{f}_{i,j+1/2}$ and $\mathbf{f}_{i+1/2,j}$ for $i > i_0$ and $j > j_0$ only.

In the multidimensional case, this is done separately in the x and y directions. The global scheme is therefore as simple as for the one dimensional case. For systems of conservation laws, the centered approximation is performed for each component of the fluxes. The ENO interpolant, when needed, must be evaluated in each characteristic field. To do this, we follow the algorithm 4.1 in [7]. That is, we evaluate the gradient of each fluxes using Roe averages for the unknown $\mathbf{A}_{j+1/2} = \frac{\partial \mathbf{f}}{\partial \mathbf{u}}|_{\mathbf{u}=\mathbf{u}_{j+1/2}^{Roe}}$, where $\mathbf{u}_{j+1/2}^{Roe} = \mathbf{R}(\mathbf{u}_{j+1}, \mathbf{u}_j)$ is the Roe average of $\mathbf{u}_j$ and $\mathbf{u}_{j+1}$ (see [10]). Denoting the left and right eigenvectors of $\mathbf{A}_{j+1/2}$ by

$\mathbf{L}^p_{j+1/2}$, $\mathbf{R}^p_{j+1/2}$, $p = 1, ..., n$, where n is the number of equations, we correct each characteristic flux $\mathbf{f}^{(p)}_{j+1/2}$ using the divided differences projected in each field $\mathbf{B}^{(p)}_{j+1/2} = \mathbf{L}^{(p)}_{j+1/2}.\mathbf{B}_{j+1/2}$, where the $\mathbf{B}_{j+1/2}$ are the successive divided differences of $\mathbf{f}_{j+1/2}$ needed in the evaluation of the interpolating polynomial. We then reconstruct each flux in the physical space $\mathbf{f}_{j+1/2} = \sum_{p=1}^{n} \mathbf{f}^{(p)}_{j+1/2}\mathbf{R}^{(p)}_{j+1/2}$. At sonic points, we replace the Roe building block by the LLF decomposition of the fluxes. Also, instead of natural divided differences of the fluxes, we apply the formula (2.11a) and (2.11b) of [7] in each field taking $\mathbf{f}^{(p)-}_{j-1/2} = \frac{1}{2}(\mathbf{f}_{j-1/2} - \lambda^{(p)}_{j-1/2}\mathbf{u}^{(p)}_{j-1/2})$, and $\mathbf{f}^{(p)+}_{j-1/2} = \frac{1}{2}(\mathbf{f}_{j-1/2} + \lambda^{(p)}_{j-1/2}\mathbf{u}^{(p)}_{j-1/2})$ where $\lambda^{(p)}_{j-1/2} = \min(\mathbf{u}^{(p)}_{j-1}, \mathbf{u}^{(p)}_{j})$ in the case of convex fluxes or genuinely nonlinear fields. For more details, see [7].

The most interesting and important part of this method consists in the way large transition areas are detected. For the basic idea, we use concepts related to a front capturing method which has been introduced in [8] in the case of combustion type problems dealing with Hamilton-Jacobi type equations and in many other applications where fronts must be located with high accuracy. For conservation laws, we will say that from time t to time $t+\Delta t$ the solution has changed considerably, if the normal at this point to the solution surface at these two different times has rotated by an angle exceeding some preset value α, $|\alpha| < |\frac{\Pi}{2}|$. In numerical computations, spatial derivatives of $\mathbf{u}^{n+i/m}_j$ are evaluated using backward or forward derivatives, so that the change of the normal is described via this formula:

$$\Delta_{\pm}\mathbf{u}^n_i \Delta_{\pm}\mathbf{u}^{n+1}_i \leq \Delta x^2\Big(-1 + \cos\alpha\sqrt{1 + \Big(\frac{\Delta_{\pm}\mathbf{u}^n_i}{\Delta x}\Big)^2}\sqrt{1 + \Big(\frac{\Delta_{\pm}\mathbf{u}^{n+1}_i}{\Delta x}\Big)^2}\Big), \tag{17}$$

where the $\pm$ tests are performed to detect fast transition locations for $\mathbf{u}_{i+1/2}$ or $\mathbf{u}_{i-1/2}$, respectively, and $\Delta_{\pm}\mathbf{u}^{n+1}_i = \pm(\mathbf{u}_{i\pm1} - \mathbf{u}_i)$ are for the forward and backward differences.

If this inequality is satisfied with the value of the parameter $-\frac{\Pi}{2} < \alpha < \frac{\Pi}{2}$ then the correction step is performed. The parameter α is introduced to restrict more general changes of the normal to the surface from different times. In the numerical examples introduced in next section, it has been necessary to tune this coefficient in order to obtain the smallest possible number of corrections without introducing oscillations in the numerical solution. If, instead, we let $\cos\alpha = 0$, more corrections must be performed. Basically, we also have to correct when an extremum point already exists at time t. This test can be avoided when larger values of the parameter $\cos\alpha$ are used. This process has greatly reduced the number of corrections when two dimensional Euler equations for compressible gas dynamics with shock-turbulence interaction has been studied. However, it has been difficult to adjust this parameter to optimize the global algorithm (computational time). Nevertheless, less than 40% of the grid points required corrections for $\cos\alpha = 0.1$. Further optimization will probably lead to even fewer corrections.

The value $\cos\alpha = 1$ implies that the filtering scheme is applied everywhere, while, $\cos\alpha = -1$ leads only to the simple basic centered difference scheme. The particular value of the parameter $\cos\alpha$, for which the right hand side of (17) is zero, implies that a change in sign of a partial derivative occured from time t to $t + \Delta t$.

We must add that no significant improvement of the solution is obtained when the value of $\cos\alpha$ approaches 1. This is probably due to the fact that the numerical error is very small in regions where the physical solution is smooth for which centered differences are used. Furthermore, as shown in the numerical examples below, there is no visible propagation of the numerical error due to the filtering step.

To conclude this section, we write down clearly the algorithm for multidimensional systems of conservation laws:

- **For** $i = 1, ..., m$ **Do**
  - **For** $I_j = 1, ..., N_j$ **Do**, $j = 1, ..., nd$, nd is the spatial dimension,
    - **Compute** $\mathbf{u}^{n+i/m}_{I_1,...,I_{nd}}$ using the basic centered difference scheme:

$$\begin{aligned}\mathbf{u}^{n+i/m}_{I_1,...,I_{nd}} &= A(\mathbf{u}^n_j, ..., \mathbf{u}^{n+(i-1)/m}_j) \\ &+ \frac{\Delta t}{\Delta x_{I_1}}(\mathbf{f}^c_{I_1+1/2,I_2,...,I_{nd}} - \mathbf{f}^c_{I_1-1/2,I_2,...,I_{nd}}) \\ &+ ... + \frac{\Delta t}{\Delta x_{I_{nd}}}(\mathbf{f}^c_{I_1,...,I_{nd}+1/2} - \mathbf{f}^c_{I1,...,I_{nd}-1/2})\end{aligned}$$

  - **End For.**
  - **For** $I_j = 1, ..., N_j$ **Do** , $j = 1, ..., nd$:
    - **Compute** the normal to the surface at time $T^{n+(i-1)/m}$ and $T^{n+i/m}$, and test whether the directional change of the normals exceeds the angle α:

$$\begin{aligned}&\frac{\Delta^{x_{I_1}}_{\pm}\mathbf{u}^{n+(i-1)/m}_{I_1,...,I_{nd}}\Delta^{x_{I_1}}_{\pm}\mathbf{u}^{n+i/m}_{I_1,...,I_{nd}}}{\Delta x^2_{I_1}} + \\ &... + \frac{\Delta^{x_{I_{nd}}}_{\pm}\mathbf{u}^{n+(i-1)/m}_{I_1,...,I_{nd}}\Delta^{x_{I_{nd}}}_{\pm}\mathbf{u}^{n+i/m}_{I_1,...,I_{nd}}}{\Delta x^2_{I_{nd}}} \\ &\leq \Big(-1 + \cos\alpha \\ &\sqrt{1 + (\frac{\Delta^{x_{I_1}}_{\pm}\mathbf{u}^{n+(i-1)/m}_{I_1,...,I_{nd}}}{\Delta x_{I_1}})^2 + ... + (\frac{\Delta^{x_{I_{nd}}}_{\pm}\mathbf{u}^{n+(i-1)/m}_{I_1,...,I_{nd}}}{\Delta x_{I_{nd}}})^2} \\ &\sqrt{1 + (\frac{\Delta^{x_{I_1}}_{\pm}\mathbf{u}^{n+i/m}_{I_1,...,I_{nd}}}{\Delta x_{I_1}})^2 + ... + (\frac{\Delta^{x_{I_{nd}}}_{\pm}\mathbf{u}^{n+i/m}_{I_1,...,I_{nd}}}{\Delta x_{I_{nd}}})^2}\Big)\end{aligned}$$

* **If** these tests are satisfied **Then** compute the ENO interpolant in each field and correct the solution.

– **End For.**

• **End For.**

The number of tests to be performed depends on the dimension of the system ($= ns$) and of space dimension ($= nd$). In general, we will have $ns * nd$ tests to be performed so that each component of the fluxes in each direction can be modified. Therefore this postprocessing may imply a high computational cost if the code is not well implemented.

3 Numerical Results

We now apply our algorithm to several test problems dealing with linear and nonlinear hyperbolic systems of conservation laws. We will focus our attention in determining precisely the order of accuracy of our filtering method and studying the propagation of the local error. We will also visualize the numerical solution when linear (contact) or nonlinear (shock) discontinuities appear and compare the computational time of the filtering method versus the associated unfiltered ENO scheme.

3.1 Example I:

As first test problem, we want to verify that our method is uniformly high order accurate even at extremum points. To do so, we consider the simple linear equation

$$u_t + u_x = 0,$$

with initial condition

$$u(x,0) = \cos 2\Pi x,$$

to be solved for $t > 0$ and $x \in [0,1]$ with periodic boundary conditions at 0 and 1. Numerically, we discretize the interval $[0,1]$ and let $x_i = i\Delta x$, for $i = 0,...,n$. The exact solution $u(x_i,t_n)$ is then approximated by pointwise values u_i^n for which we set $u_i^0 = u(x_i,0)$ from the initial condition. The exact solution is calculated by the method of characteristics so that the local numerical error and order of accuracy can be estimated. In the numerical experiments, we fixed $n = 40$ and ran the program for one period of time, e.g $t = 1$. The number of corrections per each time substep was never higher than 4. Moreover, these corrections occur at extremum points. Table 1 shows the local error at points $x_{4j} = 4j.\Delta x$. Table 2 describes the

global order of accuracy in L^1 and L^∞ norms for the (3-2), (3-4), and (3-6) filtering methods (FM).

x-location	(3-2)FM Local Error	(3-4)FM Local Error	(3-6)FM Local Error
0.	$7.15 * 10^{-2}$	$5.69 * 10^{-4}$	$1.34 * 10^{-4}$
0.1	$2.60 * 10^{-2}$	$1.70 * 10^{-4}$	$1.01 * 10^{-4}$
0.2	$1.96 * 10^{-2}$	$1.26 * 10^{-4}$	$3.36 * 10^{-5}$
0.3	$3.27 * 10^{-2}$	$6.91 * 10^{-4}$	$4.75 * 10^{-5}$
0.4	$1.25 * 10^{-2}$	$1.05 * 10^{-4}$	$1.01 * 10^{-4}$
0.5	$7.24 * 10^{-2}$	$5.59 * 10^{-4}$	$1.34 * 10^{-4}$
0.6	$2.71 * 10^{-2}$	$1.69 * 10^{-4}$	$1.01 * 10^{-4}$
0.7	$2.02 * 10^{-2}$	$1.24 * 10^{-4}$	$3.36 * 10^{-5}$
0.8	$3.31 * 10^{-2}$	$7.32 * 10^{-5}$	$4.74 * 10^{-5}$
0.9	$1.31 * 10^{-2}$	$1.10 * 10^{-4}$	$1.27 * 10^{-4}$
1.	$6.29 * 10^{-2}$	$5.58 * 10^{-4}$	$1.32 * 10^{-4}$

Table 1.

Scheme	L^1-norm	L^∞-norm
(3-2)FM	2.08	1.75
(3-4)FM	4.11	3.46
(3-6)FM	3.46	3.26

Table 2.

These results are indeed in agreement with what we should expect. Uniform high order is preserved even at corrected points.

3.2 Example II:

We now extend Example I to two dimensions. As described in the algorithm, a dimension by dimension approach is used to solve the linear equation

$$\begin{aligned} u_t + u_x + u_y &= 0, \\ u(x,y,0) &= \cos 2\Pi(x+y) \text{ and } 0 \le x,y \le 1, \end{aligned}$$

and again, periodic boundary conditions in both x and y variables are assumed. We discretize the square domain $\Omega = [0,1] \times [0,1]$ and denote by $A_{i,j}$ the vertices of coordinates $x_i = i\Delta x$, and $y_j = j\Delta y$, for $i = 0, ..., n$ and $j = 0, ..., m$. We choose $n \neq m$ so that the problem is really two dimensional. The exact solution of this equation can be easily calculated using the change of variables $\xi = x + y$ leading to a one dimensional problem. Hence, the exact solution is merely

$$u(x,y,t) = \cos(2\Pi(x+y-t)).$$

Numerically, we use $m = 40$, and $n = 30$ so that $\Delta x \neq \Delta y$. The figures (2.1), (2.2), and (2.3) visualize the local error for the (3-2), (3-4), and (3-6) FM at different sections $x = 0.2, 0.4, 0.6, 0.8$. The local error is highest at extremum points, particularly for the second order method which is globally TVD. However, table 3 shows that the global order of

accuracy in L^1 and L^∞ norms is uniform. Moreover, table 4 shows that the order of accuracy for the filtering method does not really depend on the value of the parameter α. However, the number of corrections increases as $\cos\alpha$ approaches 1 for which the full ENO method is performed.

Scheme	L^1-norm	L^∞-norm
(3-2)FM	2.00	1.45
(3-4)FM	3.81	3.35
(3-6)FM	3.06	3.11

Table 3.

$\cos\alpha$	# of corrections	L^1-norm	L^∞-norm	CPU time
0.1 (3-4)FM	84	3.62	3.01	13.23
0.9 (3-4)FM	253	3.61	3.02	13.45
1.0 (3-4)RM	1681	3.91	3.15	36.41

Table 4.

From this table, the computational cost is reduced by a factor of almost 3 as the (3-4)FM is used ($\cos\alpha \leq 0.9$) versus the full ENO (3-4)RF method ($\cos\alpha = 1.$). Also, no significant gain in the order of accuracy is obtained when the full (3-4)RF method is used.

3.3 Example III:

In this example, we study the behavior of the filtering method to nonlinear equations. As simple case, we consider Burgers' equation

$$u_t + \left(\frac{u^2}{2}\right)_x = 0,$$

with initial condition

$$u(x,0) = \cos 2\Pi x.$$

We again take periodic boundary conditions on the interval $[0,1]$. The solution becomes discontinuous at time $t = \frac{1}{2\Pi}$ at the steady location $x = 0.25$. The domain Ω is discretized and the grid points are denoted by x_i, for $i = 0,...,n$, and let $n = 40$ in the numerical experiments. The exact solution is approximated by using Newton's method whenever the solution is smooth. The local error at each grid points is plotted for the (3-2), (3-4), and (3-6)FM in the figures 3.1, 3.2, and 3.3. Due to the centered differences which are performed in the basic scheme, the local error propagates symmetrically with respect to inflection points. Nevertheless, the table 5 shows that the order of accuracy of the filtering method is still uniform. These calculations were performed at time $t = 0.1$ with a CFL coefficient $\lambda = 1.$.

Scheme	L^1-norm	L^∞-norm
(3-2)FM	1.95	1.40
(3-4)FM	3.74	3.69
(3-6)FM	3.50	3.47

Table 5.

At time $t = \frac{1}{2\Pi}$ the shock wave appears. The shape of the solution is shown in the set of figures 3.4, 3.5, and 3.6. No spurious oscillations can be detected from these plots. The number of corrections was never higher than 8 whatever the number of grid points. Again no visible change in the solution occurs when different values of α are considered. In this numerical example, $\cos\alpha = 0.1$ and the extremum test was enforced.

Using Burgers' equation again, we want to test whether our method is stable. We consider the Riemann problem with $u_l = +1$, and $u_r = -1$ and run our program using centered differences only on a 40 point grid with a CFL number of 0.5 up to $t = 3$. The figure (3.7.1) visualizes the solution at this time. Large oscillations can be seen up to the boundaries. Taking the final solution of the previous problem as initial condition and using the (3-2)FM with the parameter $\cos\alpha = 0.99$, we obtain the steady solution $u_l = +1$, $u_r = -1$ (figure (3.7.2). Moreover, the test for an extrema was not used. This shows that our filtering method acts like a viscosity method in regions of smoothness even as the parameter $\cos\alpha$ approaches 1. However, the number of corrections occured at only 10 grid points on the average.

Finally, we want to test whether our method works for nonconvex fluxes and for initial conditions solving a Riemann problem having an expansion shock. First of all, we considered Burgers' equation with the initial condition $u_l = -2$ for $x \leq 0.$, and $u_r = 2$ for $x > 0$. The centered differences of the basic scheme let the initial expansion shock unchanged. However, by adding a small perturbation of amplitude $\varepsilon = 10^{-3}$ to the initial condition, the numerical solution does not violate the entropy condition and does tend for long time to the stationary solution of the problem, i.e $u = 0$ (see the figure 3.7.3). In the other hand, we ran our program with a nonconvex flux $f(u) = \frac{(u^2-1)(u^2-4)}{4}$, with the initial condition $u_l = 2$, and $u_r = -2$ in each side of $x = 0$. Again, if no perturbation is added to this initial condition, the centered differences let the solution remain unchanged. The results with a small initial noise added to the initial condition are displayed in the figure 3.7.4. The results are in agreement with those presented in [6].

In order to avoid the risk of developing an expansion shock, we may implement another test that checks whether $\lambda_i^L < 0 < \lambda_i^R$, where $\lambda_i^{L,R}$ are the eigenvalues of $\nabla_{\mathbf{u}}\mathbf{f}$ in each side of the grid point x_i in the genuinely nonlinear fields (for non convex fields, we correct at all sonic points). If this test is satisfied then we correct the numerical solution using the filter.

3.4 Example IV:

In this example, we extend example III to two space dimensions by taking the two dimensional Burgers' equation

$$u_t + \left(\frac{u^2}{2}\right)_x + \left(\frac{u^2}{2}\right)_y = 0,$$

with the same initial condition as in example II and 1-periodic boundary conditions in both x and y variables. The square domain is discretized as in example II with $m = 40$ and $n = 30$. The shock discontinuity occurs at time $t = \frac{1}{2\Pi}$ at the steady location $x + y = 0.25$. We study the propagation of the local error at time $t = 0.1$ at different sections $x = 0.2, 0.4, 0.6, 0.8$. Again the error is symmetric with respect to inflection points, see the figures (4.1), (4.2), and (4.3). Moreover, the error is distributed within more grid points as the order of the space discretization increases. This is in agreement with the fact that the width of the stencil increases with the order of accuracy. Table 6 describes the order of accuracy in L^1 and L^∞ norms for the (3-2), (3-4), and (3-6) filtering methods. Table 7 discusses the order of accuracy in these norms for different values of the parameter $\cos\alpha$ for the (3,4)FM. We also compare the CPU time of the (3-4)FM method versus the (3-4)LLF ENO method.

Scheme	L^1-norm	L^∞-norm
(3-2)FM	1.86	1.64
(3-4)FM	3.47	3.19
(3-6)FM	4.13	3.49

Table 6.

$\cos\alpha$	# of corrections	L^1-norm	L^∞-norm	CPU time
0.1 (3-4)FM	126	3.25	3.19	15.51
0.9 (3-4)FM	211	3.24	3.25	18.32
1.0 (3-4)LLF	1681	3.33	3.38	39.71

Table 7.

The CPU time is again reduced by a factor of almost 3 when the filtering method is performed.

At time $t = \frac{1}{2\Pi}$ the shock discontinuity occurs and is visualized in the figure 4.4. A sharp transition is obtained and no spurious oscillations can be seen. Away from the shock the local order of accuracy is preserved showing that no propagation of error starting at the shock location pollutes the smooth part of the numerical solution. Moreover, the formal orders of accuracy are approximately the same as those given in the table 6.

3.5 Examples V:

In these examples, we study the smearing of contact discontinuities for one and two dimensional problems. To illustrate this fact and study the behavior of the filtering method in such cases, we consider the linear equations given in examples I and II with different initial conditions:

$$\begin{aligned} u(x,0) &= \cos(2\Pi x) \pm 0.5 \text{ If } 0.25 \leq x \text{ or } x \geq 0.75 \\ \text{and } u(x,0) &= 0 \text{ otherwise,} \\ u(x,y,0) &= \cos(2\Pi(x+y)) + 2 \text{ If } 0.25 \leq x, y \leq 0.75 \\ \text{and } u(x,y,0) &= 0 \text{ otherwise,} \end{aligned}$$

for one and two dimensional problems, respectively. In the numerical experiments we let $n = 40$ and study the numerical solution at time $t = 1.1$ with a CFL coefficient $\lambda = 0.5$.

For the one dimensional problem, the transition is plotted in the set of figures (5.1.1), (5.1.2), and (5.1.3) for the (3-2), (3-4), and (3-6) FM, respectively. In all cases, the jump transition is localized within a few mesh points. However, as expected, the numerical solution near the contact discontinuities is better approximated for the highest (sixth) order method. In particular, the small oscillations that appear near the contact discontinuities for the second and fourth order methods, are no longer there for the sixth order method. Also, the small oscillations will disappear as soon as the value of the parameter $\cos\alpha$ is not less than 0.9. Moreover, the transition from the upper to the lower parts of the numerical solution is better resolved for high order methods (fourth and sixth order), whereas the transition tends to be smeared for the second order method.

In the next set of plots, the two dimensional problem is investigated. The results are shown in the figures (5.2.1), (5.3.1), (5.4.1) where the solution wave is plotted at time $t = 1.1$. The transition from the zero plateau to the cosine wave is smeared within three to four meshes in each vertical and horizontal sides and probably more at each of the four corners, see the figures (5.2.2), (5.3.2), and (5.4.2). Moreover, the maximum error occurs at the bottom left and the top right corners. Also, the local error is very smooth along each sides and "discontinuous" about these two corners. This agrees with the one dimensional results in which the smearing of the contact discontinuities happens to be more important in the upper or lower part of the cosine wave.

3.6 Examples VI:

The last examples are devoted to extend our filtering method to systems of conservation laws. We consider the compressible Euler equations for gas dynamics introduced during the introduction. The Euler equations (1) and (4) are studied with these sets of initial conditions:

- **One dimensional Euler:**
  - We consider the initial condition given in the example 8 of [7]. That is, we take:

$$\rho = 3.857143, q = 2.629369, p = 10.3333333 \text{ when } x < -4$$
$$\rho = 1 + \varepsilon \sin 5x, q = 0, p = 1. \text{ when } x \geq -4$$

  - If $\varepsilon = 0.$ we get a pure Mach 3 shock moving to the right. Following [7] in example 8, we take $\varepsilon = 0.2$ in the numerical experiment.

- **Two dimensional Euler:**

  - We consider the initial condition given in the example 9 of [7]. That is, we consider a Mach 8 shock located at $x = -1$ moving to the right into the state with

$$\begin{aligned} p_r &= 1, \\ \rho_r &= 1, \\ v_x &= -\frac{c_r}{p_r} \sin\theta_r \cos\left(x k_r cos\theta_r + y k_r \sin\theta_r\right), \\ v_y &= \frac{c_r}{p_r} \cos\theta_r \cos\left(x k_r cos\theta_r + y k_r \sin\theta_r\right), \end{aligned}$$

    where $k_r = \frac{2}{\Pi}$ and $\theta_r = \frac{\Pi}{6}$. In order to have positive pressure during the calculations, we used a parameter $\cos\alpha = 0.1 \rightarrow \alpha \simeq 0.9\frac{\Pi}{2}$.

The results for the one dimensional problem are shown in the set of figures (6.1.1), (6.1.2), and (6.1.3). The desired, physical, oscillations near the shock transition which are parts of the exact solution are particularly visible for the sixth order method. The second order and fourth order method give a fairly good representation of the expected solution. The number of corrections was about 25% for 200 grid points and a CFL coefficient $\lambda = 0.8$.

The results for the two dimensional shock-turbulence problem are plotted in the set of figures (6.2.1), (6.2.2), (6.3.1), (6.3.2), in which the pressure and density field are displayed. As comparison, similar results have been obtained in Using this value of the parameter $\cos\alpha$, the number of corrections was approximatively 40% for a 80×60 grid with a CFL coefficient of 0.5. In this experiment, the (3-2) and (3-4)FM have been used. 200 time iterations have been necessary to reach the final time $t = 0.2$. 70% of the global computational time was used to evaluate the ENO interpolating polynomials during the filtering step, leading to a reduction of a factor of 2 of the computational time when the filtering method is used.

4 Final Remarks and Conclusions

The main conclusion concerns the number of corrections which has always been less than 10 to 25% for all these examples, except for the two dimensional shock-turbulence problem in which a very complicated structure of the flow appears. Therefore, the computational cost for these type of problems is reduced by a factor of almost 3. Moreover, this factor is quite significant when very high order methods are implemented. In particular, the use of sixth order method for the one dimensional Euler equations leads to a very accurate approximate solution. An important remark is that it would be possible in the near future

to implement other high resolution techniques together with this type of filtering method, e.g subcell resolution [4].

In the near future, this type of method will be implemented in for different type of problems involving Hamilton-Jacobi equations. Moreover, a similar approach for more general domains using finite element triangulations is under investigation. So far, by appropriate linear combination using the basis functions of all triangles around a vertex, it has been possible to construct a second order method in space and correct the oscillating points by using a first order Godunov type filter. Higher order filtering method are also under investigation. ENO schemes for Hamilton-Jacobi equations in Cartesian coordinates were developed in [8] and in [9].

References

[1] Engquist B., P. Lotstedt, B. Sjogreen," Non linear filters for efficient shock computations", Mathematics of Computations, **52** (1989), pp509-537.

[2] Harten A., B. Engquist, S. Osher, S. Chakravarthy, " Uniformly high order accurate essentially non-oscillatory scheme III", J. Comp Physics, **71** (1987), pp 231-323.

[3] Harten A., "High resolution schemes for hyperbolic conservation laws", J. Comp Physics, **44**, 1 (1983), pp357-393.

[4] Harten A., "ENO scheme with subcell resolution", J. Comp Physics, **83** (1989), pp148-184.

[5] Kreiss H.-O., J. Oliger, "Methods for the approximate solution of time dependent problems", GARP publications, series, # 10 (1973).

[6] Osher S., C-W. Shu, "Efficient implementation of Essentially Non-Oscillatory Shock Capturing Schemes", J. Comp Physics, **77** (1988), pp 439-471.

[7] Osher S., C-W. Shu, "Efficient implementation of Essentially Non-Oscillatory Shock Capturing Schemes II", J. Comp Physics, **83** (1989), pp 32-78.

[8] Osher S., J. Sethian ,"Front propagation with curvature dependent speed: algorithms based on Hamilton-Jacobi formulations", J. Comp Physics, **79** (1988),pp 12-49.

[9] Osher S., C-W. Shu , "High order essentially non-oscillatory schemes for Hamilton-Jacobi equations", UCLA CAM Report, # 89-32, (1989).

[10] Roe P.L., "Approximate Riemann solvers, parameter vectors, and difference schemes", J. Comp Physics, **43** (1981), pp 357-372.

[11] Sjogreen B. , "Efficient computation of shock wave solutions to hyperbolic conservation laws", PhD dissertation in Numerical Analysis, (April 1989), Uppsala University, Sweden, Department of Computer Science.

[12] Yang H., Ph.D. Thesis, UCLA Math Dept., 1988.

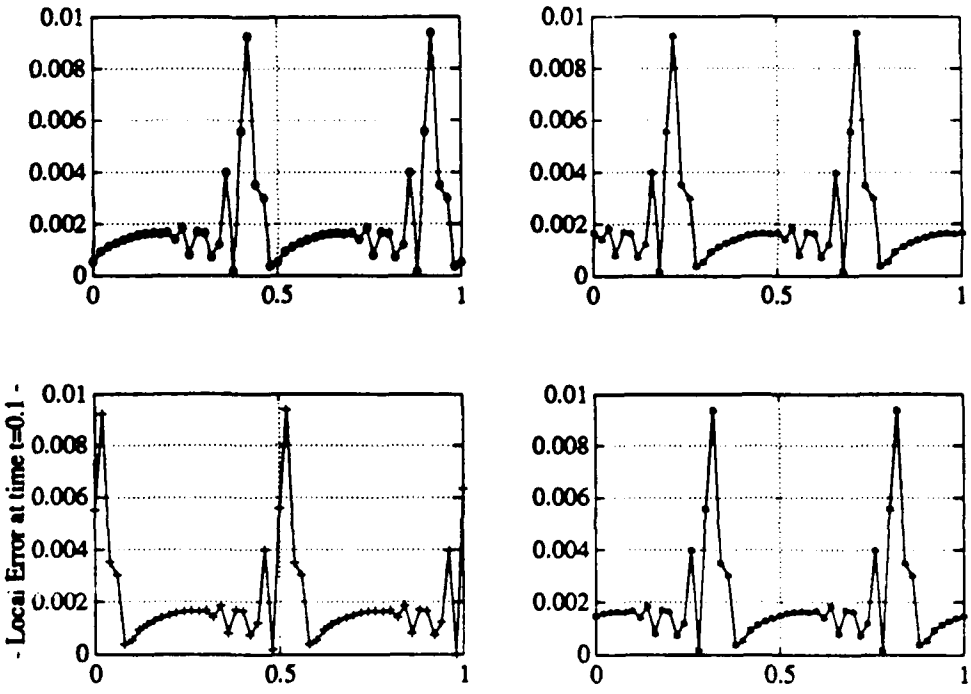
[13] Zang T., M. Hussaini, D. Bushnell, **AIAA J. 22**, 13 (1984).

[14] Zang T., M. Kopriva, M. Hussaini, Pseudospectral Calculation of Shock Turbulence Interactions **ICASE Report 83-14** (1983)

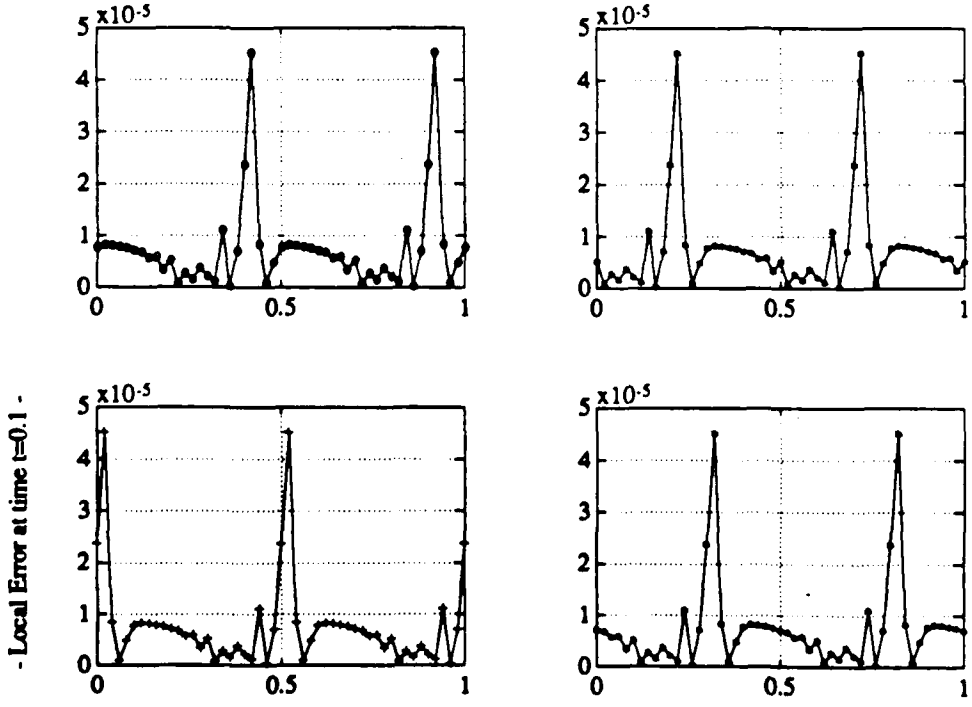
Figure Captions

- (2.1) (3-2)FM: $x = 0.2, 0.4, 0.6, 0.8$ sections, Local Error $.10^{-2}$ For 2D linear Problem (Example II).
- (2.2) (3-4)FM: $x = 0.2, 0.4, 0.6, 0.8$ sections, Local Error $.10^{-5}$ For 2D linear Problem (Example II).
- (2.3) (3-6)FM: $x = 0.2, 0.4, 0.6, 0.8$ sections, Local Error $.10^{-5}$ For 2D linear Problem (Example II).
- (3.1) (3-2)FM: Local Error $.10^{-3}$ For Burgers' Equation ($t = 0.1$).
- (3.2) (3-4)FM: Local Error $.10^{-5}$ For Burgers' Equation ($t = 0.1$).
- (3.3) (3-6)FM: Local Error $.10^{-6}$ For Burgers' Equation ($t = 0.1$).
- (3.4) (3-2)FM: Shock Transition for Burgers' Equation at time $t = \frac{1}{2\Pi}$.
- (3.5) (3-4)FM: Shock Transition for Burgers' Equation at time $t = \frac{1}{2\Pi}$.
- (3.6) (3-6)FM: Shock Transition for Burgers' Equation at time $t = \frac{1}{2\Pi}$.
- (3.7.1) (3-2)CD (Centered Differences): Solution Wave of Burgers' Equation at time $t = 1.$ with $+1, -1$ Initial Condition.
- (3.7.2) (3-2)FM: Solution Wave of Burgers' Equation at time $t = 1.$ with Initial Condition of Figure (3.7.1).
- (3.7.3) (3-4)FM: Solution Wave of Burgers' Equation at time $t = 1., 2., 3.$ with Initial Condition $u_l = -2, u_r = 2$ plus $\pm 10^{-3}$ noise.
- (3.7.4) (3-4)FM: Solution Wave of nonconvex nonlinear Equation $u_t + \frac{(u^2-1)(u^2-4)}{4}_x = 0$ at time $t = 0.2, 0.4, 0.8$ with Initial Condition $u_l = 2, u_r = -2$ plus $\pm 10^{-3}$ noise.
- (4.1) (3-2)FM: Local Error For 2D Burgers' Equation ($t = 0.1$).
- (4.2) (3-4)FM: Local Error For 2D Burgers' Equation ($t = 0.1$).
- (4.3) (3-6)FM: Local Error For 2D Burgers' Equation ($t = 0.1$).
- (4.4) (3-2)FM: Shock Transition For 2D Burgers' Equation at time $t = \frac{1}{2\Pi}$.
- (5.1.1) (3-2)FM: Linear 1D Equation - Contact Discontinuity at time $t = 1.1$.

- (5.1.2) (3-4)FM: Linear 1D Equation - Contact Discontinuity at time $t = 1.1$.
- (5.1.3) (3-6)FM: Linear 1D Equation - Contact Discontinuity at time $t = 1.1$.
- (5.2.1) (3-2)FM: Linear 2D Equation - Contact Discontinuity at time $t = 1.1$.
- (5.2.2) (3-2)FM: Linear 2D Equation - Error Plot and Contour - Contact Discontinuity at time $t = 1.1$.
- (5.3.1) (3-4)FM: Linear 2D Equation - Contact Discontinuity at time $t = 1.1$.
- (5.3.2) (3-4)FM: Linear 2D Equation - Error Plot and Contour - Contact Discontinuity at time $t = 1.1$.
- (5.4.1) (3-6)FM: Linear 2D Equation - Contact Discontinuity at time $t = 1.1$.
- (5.4.2) (3-6)FM: Linear 2D Equation - Error Plot and Contour - Contact Discontinuity at time $t = 1.1$.
- (6.1.1) (3-2)FM: 1D Euler Equations $\varepsilon = 0.2$.
- (6.1.2) (3-4)FM: 1D Euler Equations $\varepsilon = 0.2$.
- (6.1.3) (3-6)FM: 1D Euler Equations $\varepsilon = 0.2$.
- (6.2.1) (3-2)FM: 2D Euler Equations - Shock-Turbulence Interaction - Pressure field $t = 0.2$.
- (6.2.2) (3-2)FM: 2D Euler Equations - Shock-Turbulence Interaction - Density field $t = 0.2$.
- (6.3.1) (3-4)FM: 2D Euler Equations - Shock Turbulence Interaction - Pressure field $t = 0.2$.
- (6.3.2) (3-4)FM: 2D Euler Equations - Shock Turbulence Interaction - Density field $t = 0.2$.

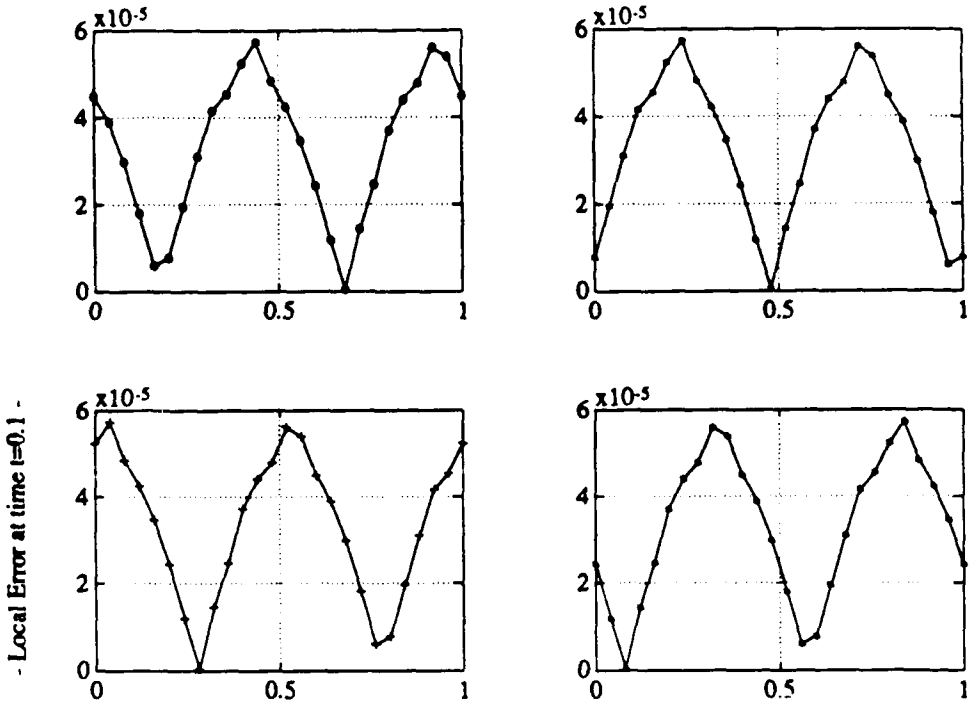


- (3,2) CD scheme + uniform ENO-Roe-2 filter , figure 2.1 -



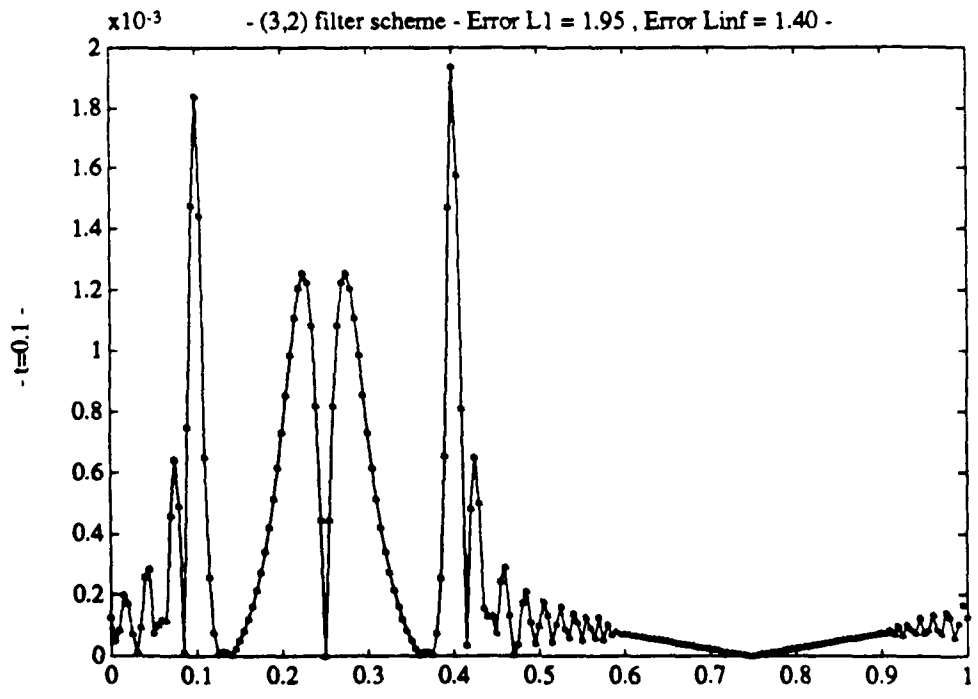
- (3,4) CD scheme + uniform ENO-Roe-4 filter -

- figure 2.2 -

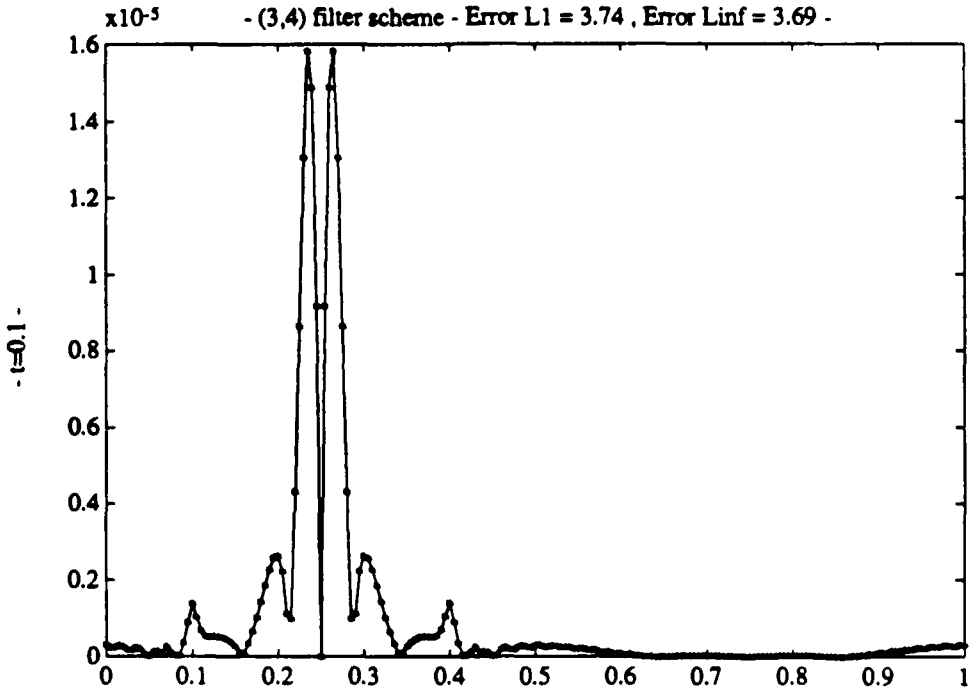


- (3,6) CD scheme + uniform ENO-Roe-6 filter -

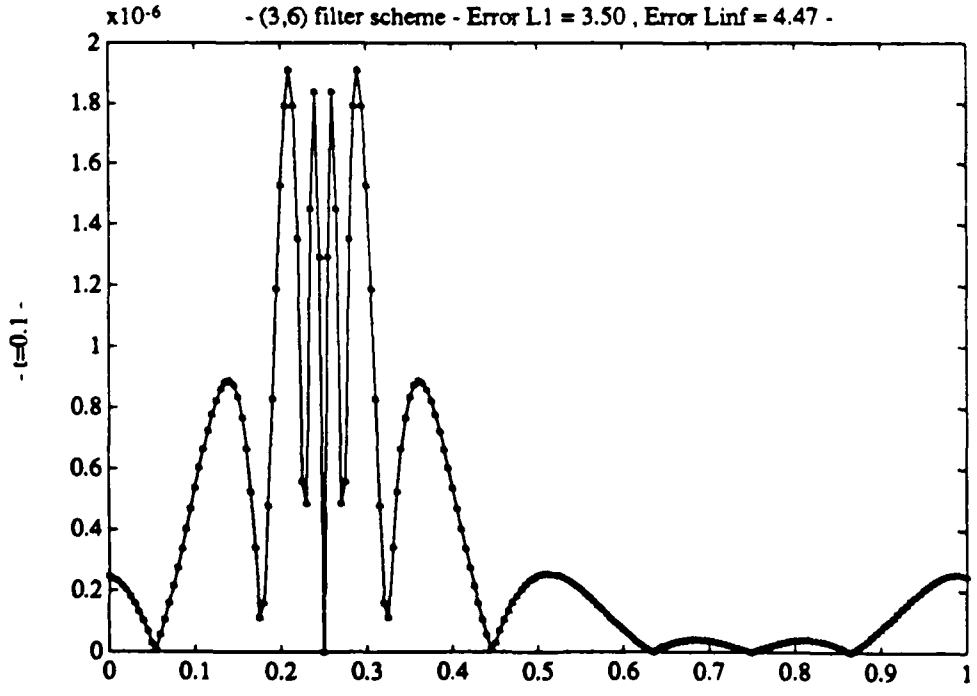
- figure 2.3 -



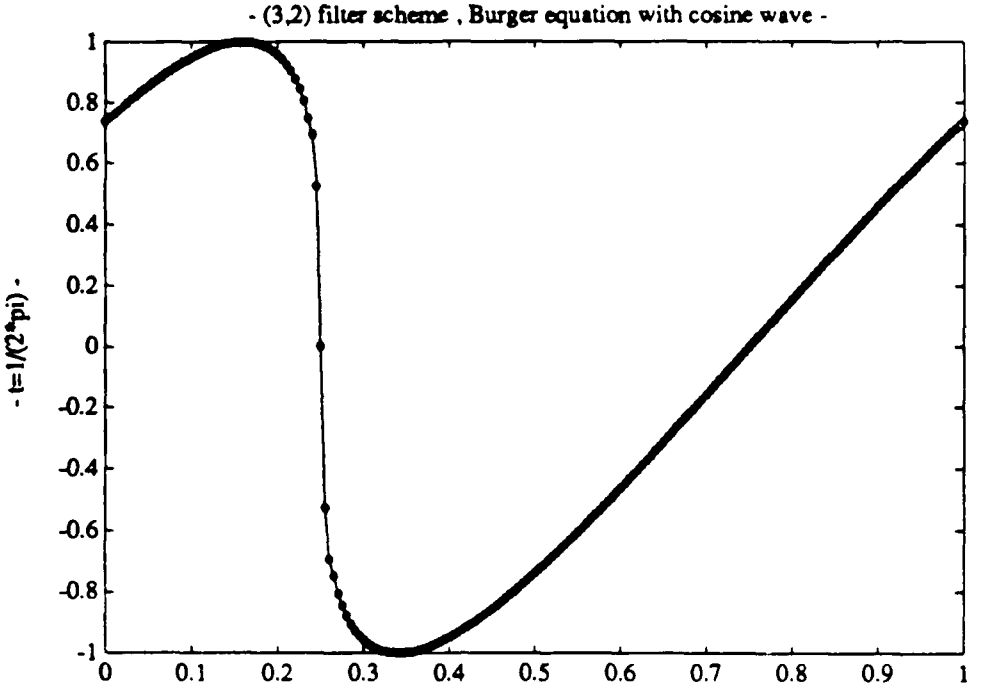
- Pointwise Error - Figure 3.1 -



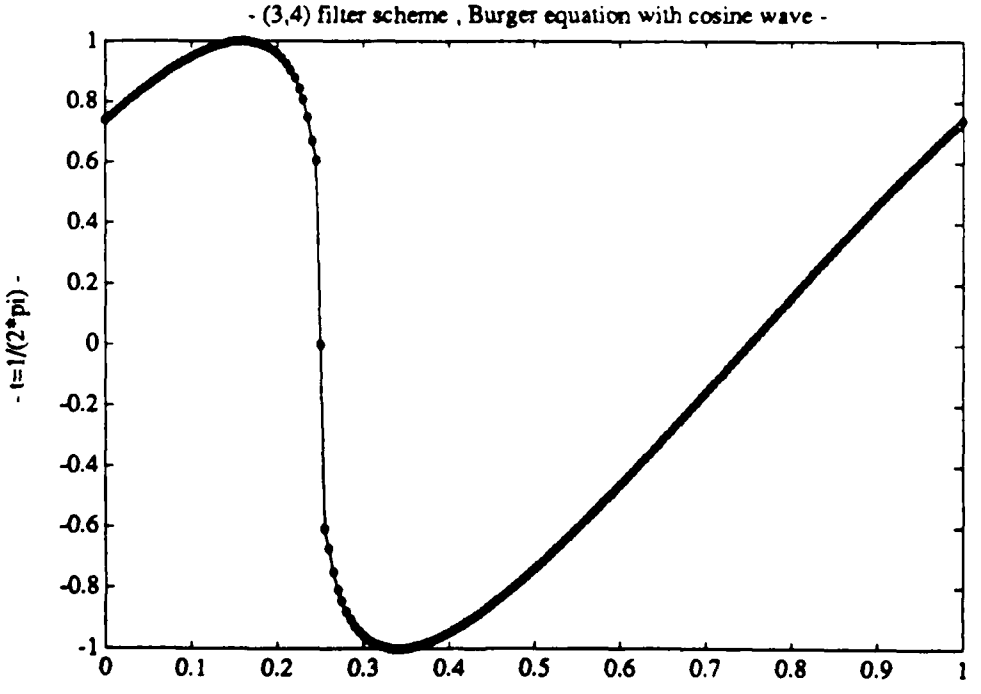
- Pointwise Error - Figure 3.2 -



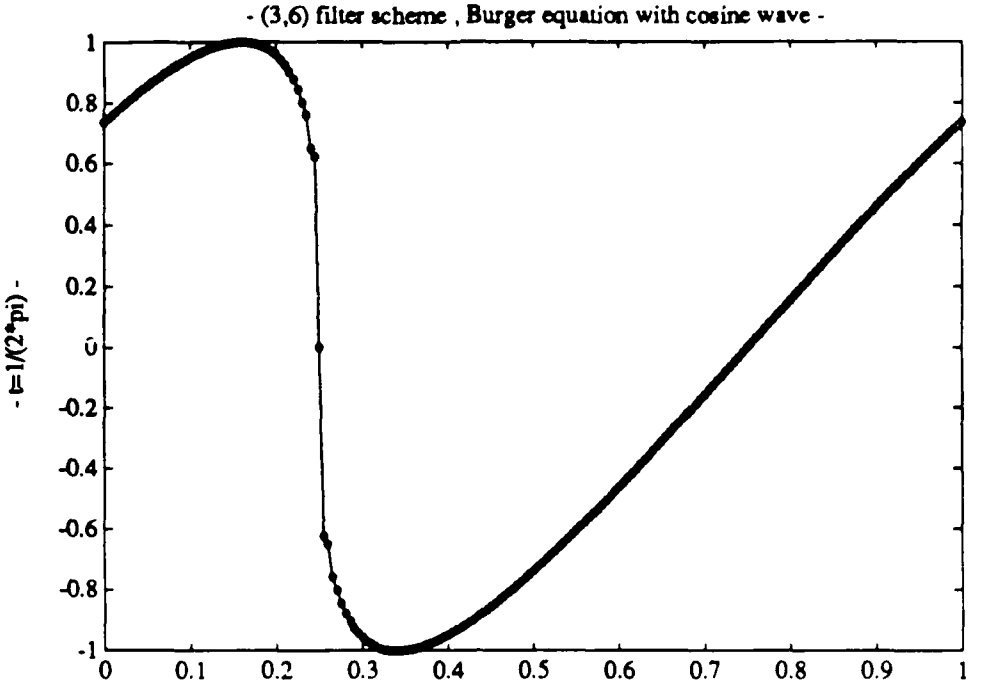
- Pointwise Error - Figure 3.3 -



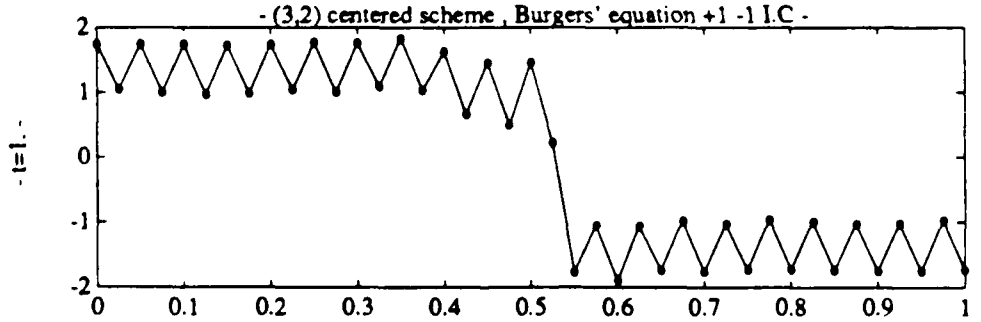
- Solution wave - Figure 3.4 -



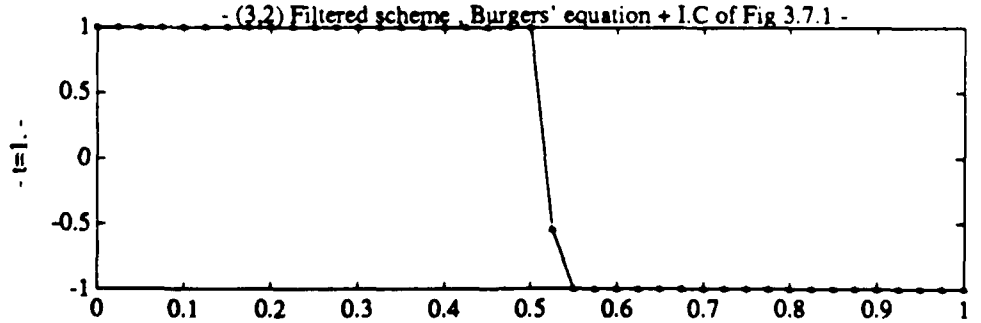
- Solution wave - Figure 3.5 -



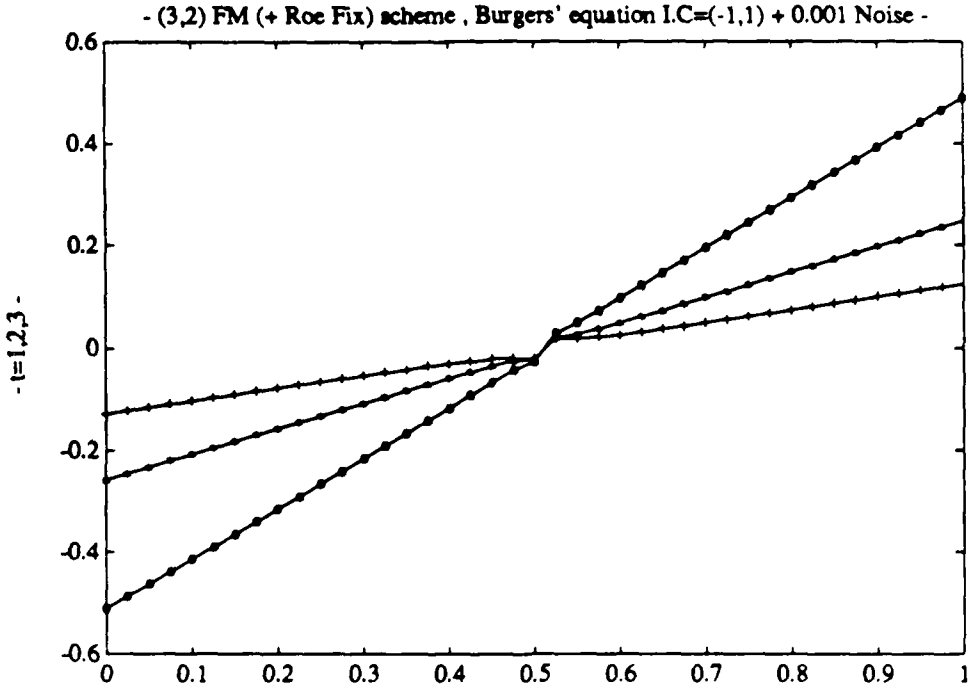
- Solution wave - Figure 3.6 -



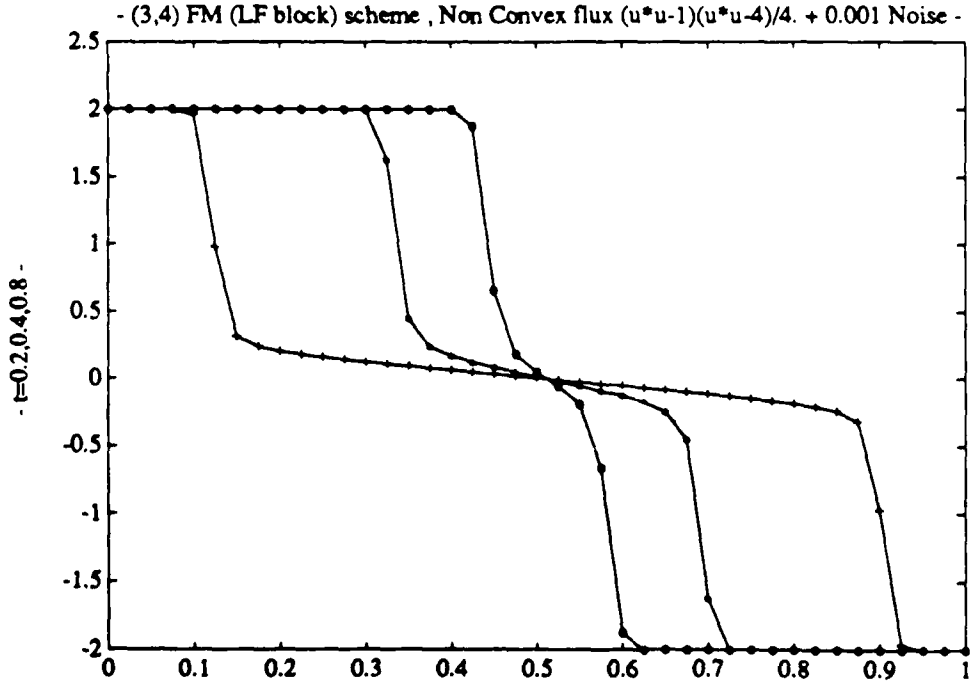
- Solution wave - Figure 3.7.1 -



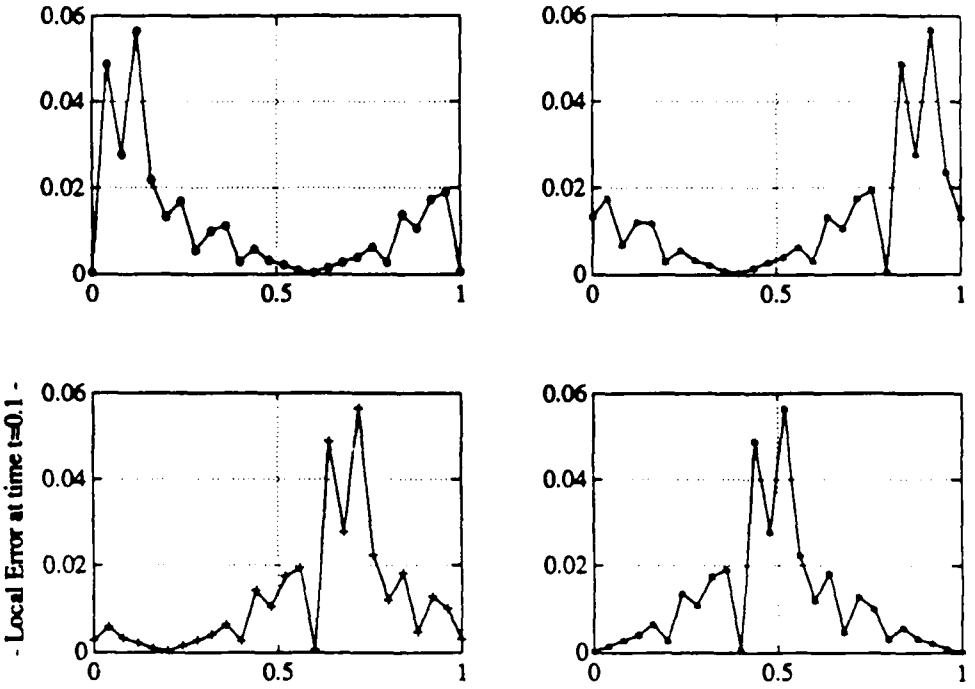
- Solution wave - Figure 3.7.2 -



- Solution wave - Figure 3.7.3 -

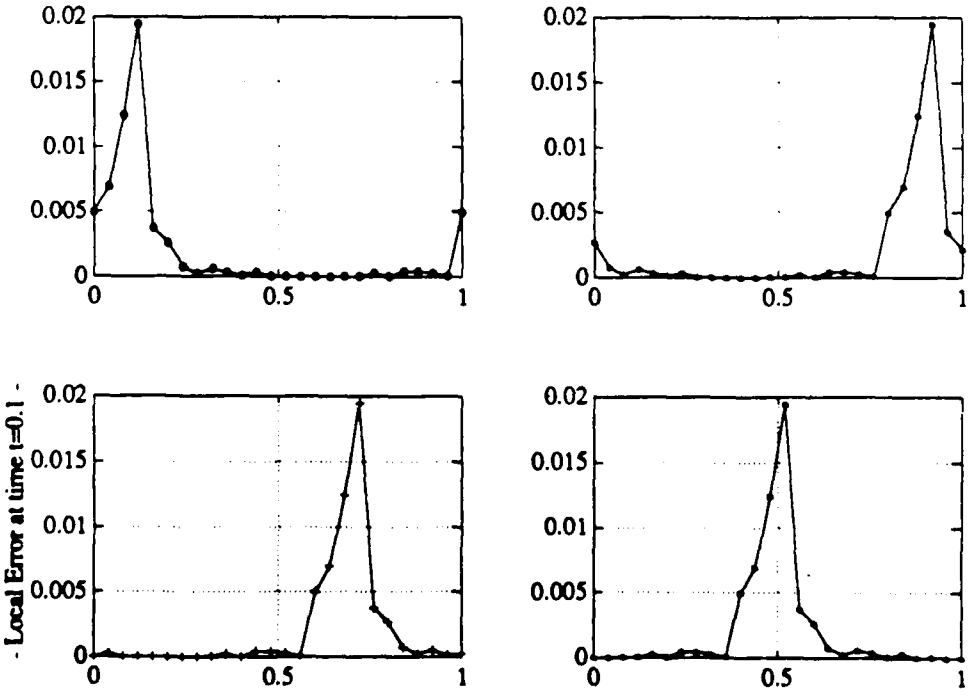


- Solution wave - Figure 3.7.4 -



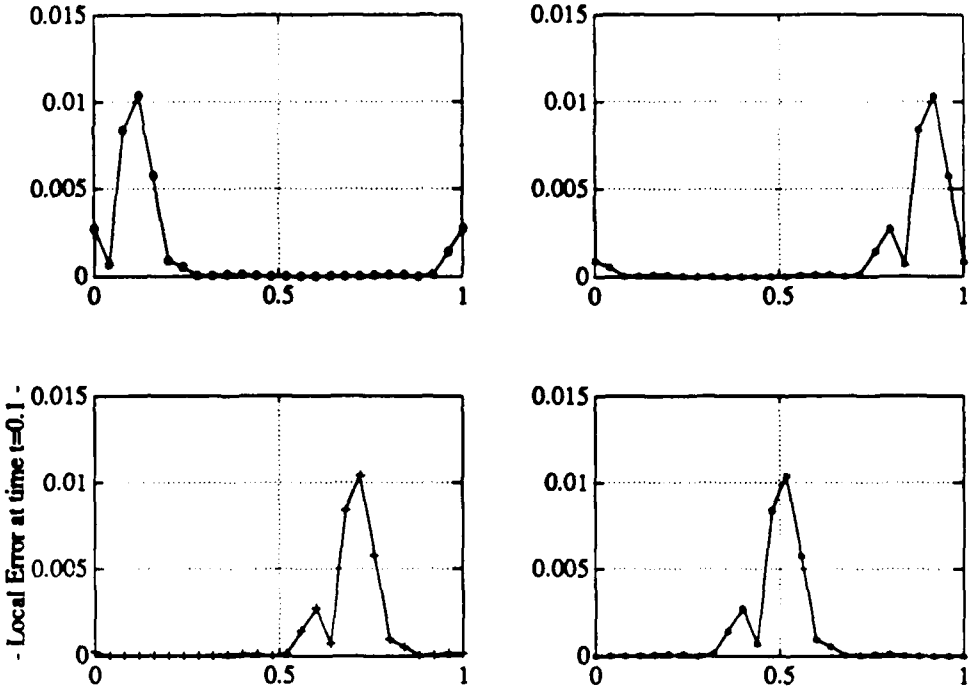
- (3,2) CD scheme + uniform ENO-Roe-2 filter -

- figure 4.1 -



- (3,4) CD scheme + uniform ENO-Roe-4 filter -

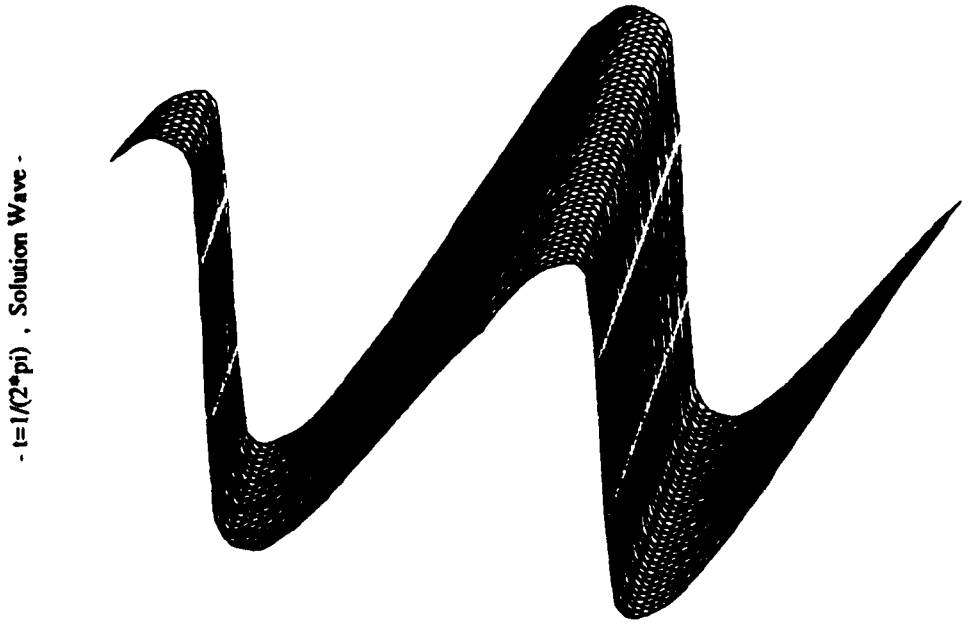
- figure 4.2 -



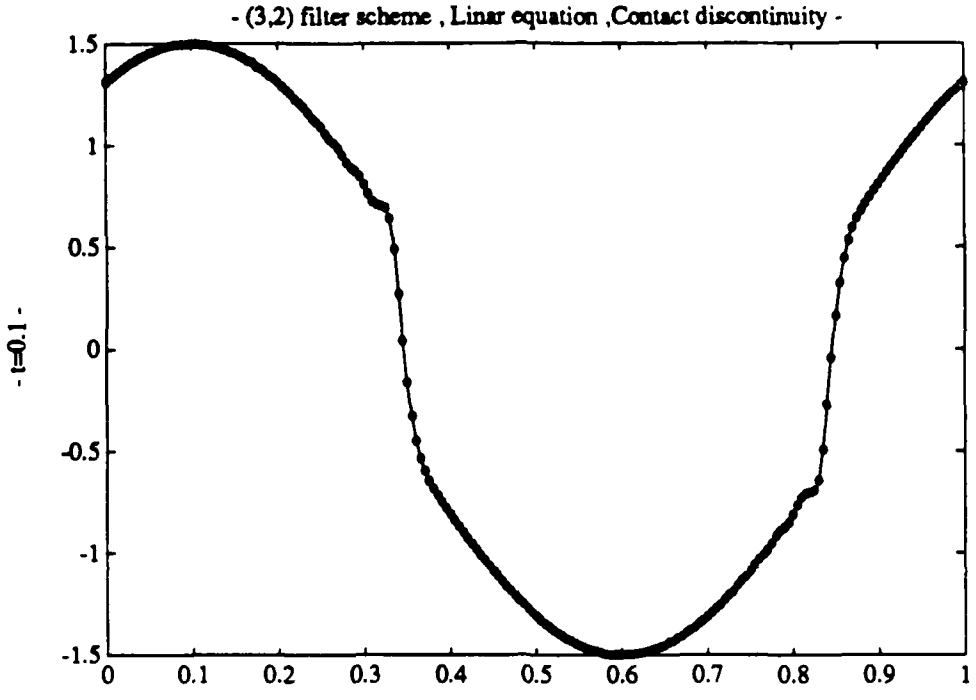
- (3,6) CD scheme + uniform ENO-Roe-6 filter -

- figure 4.3 -

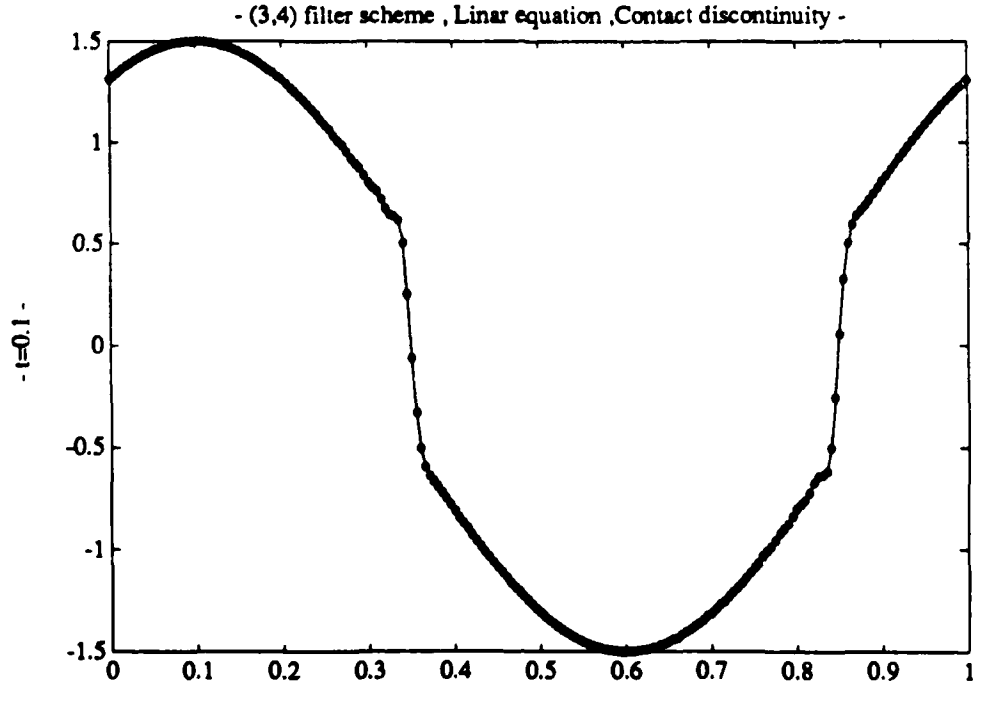
- (3,2) Filter scheme , Burger Equation Ut+(U*U/2)x+(U*U/2)y=0 IC = COS(X+Y) -



- figure 4.4 -

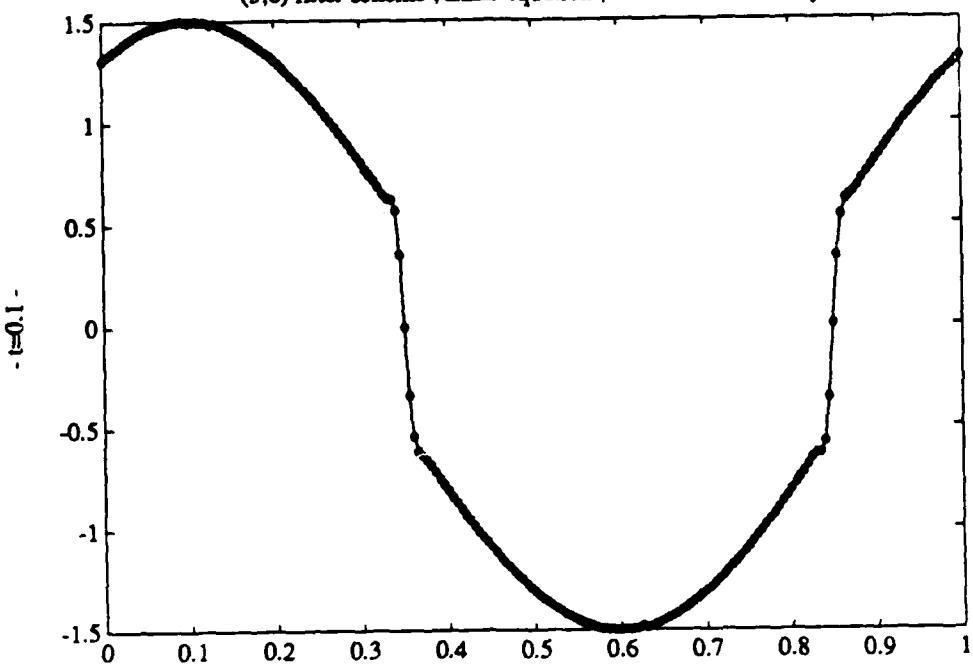


- Solution wave - Figure 5.1.1 -



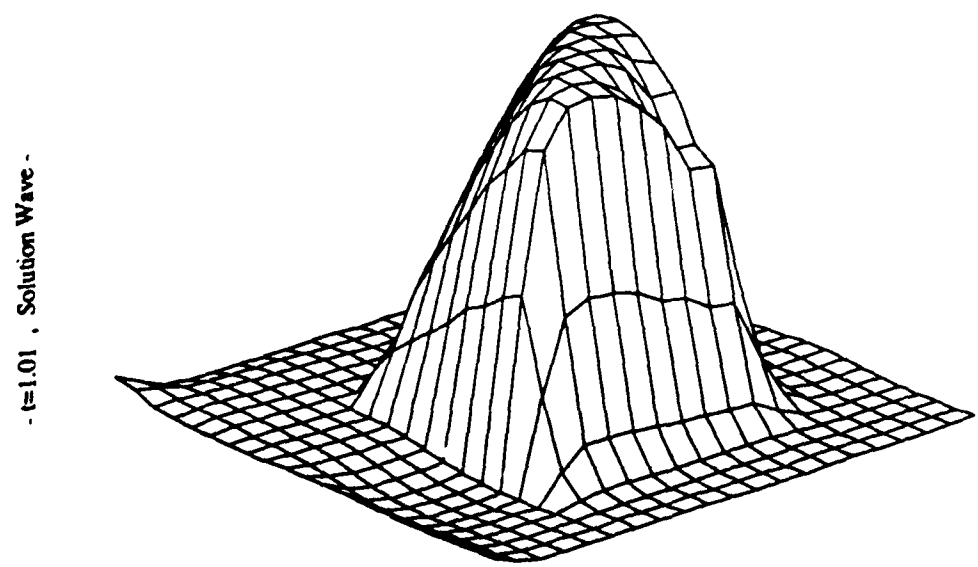
- Solution wave - Figure 5.1.2 -

- (3,6) filter scheme , Linar equation ,Contact discontinuity -

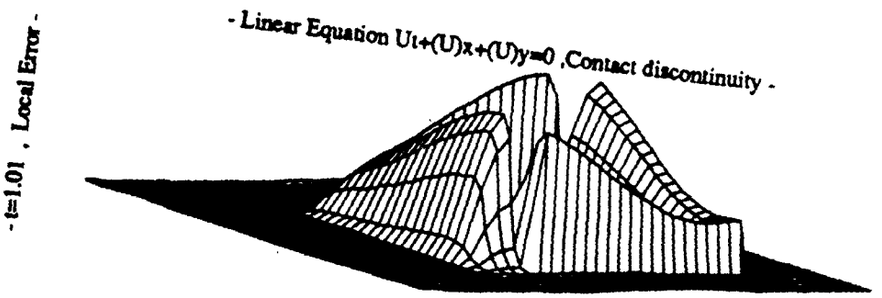


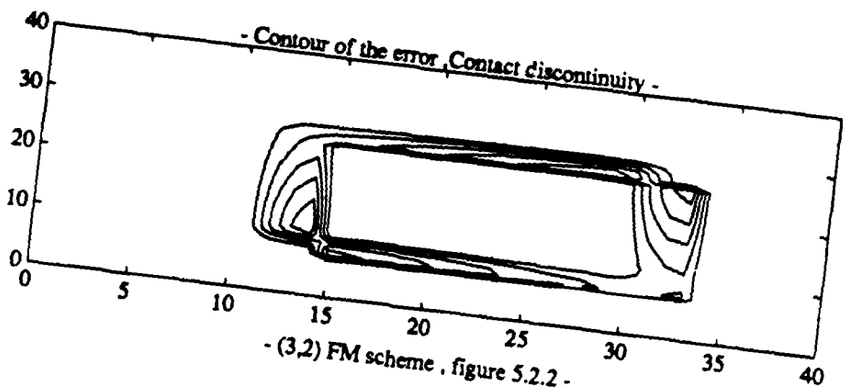
- Solution wave - Figure 5.1.3 -

- Linear Equation Ut+(U)x+(U)y=0 ,Contact discontinuity -



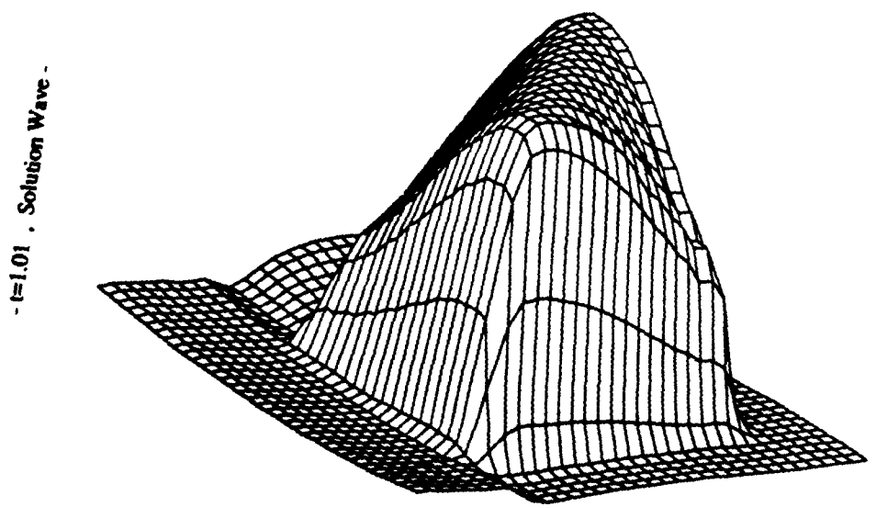
- (3,2) FM scheme , figure 5.2.1 -



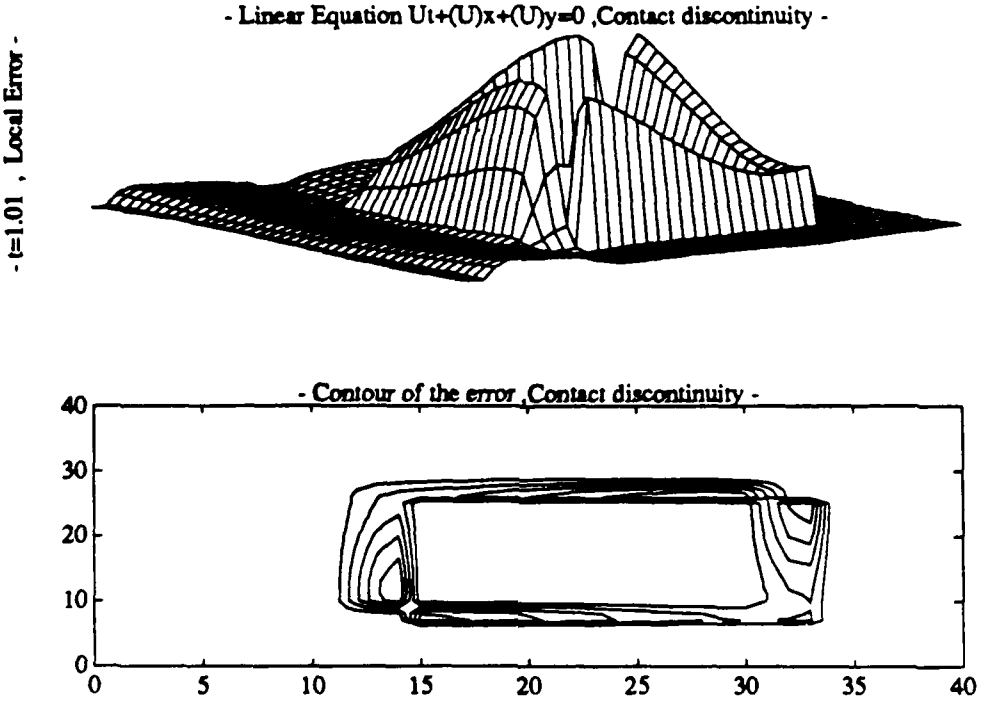


- (3,2) FM scheme , figure 5.2.2 -

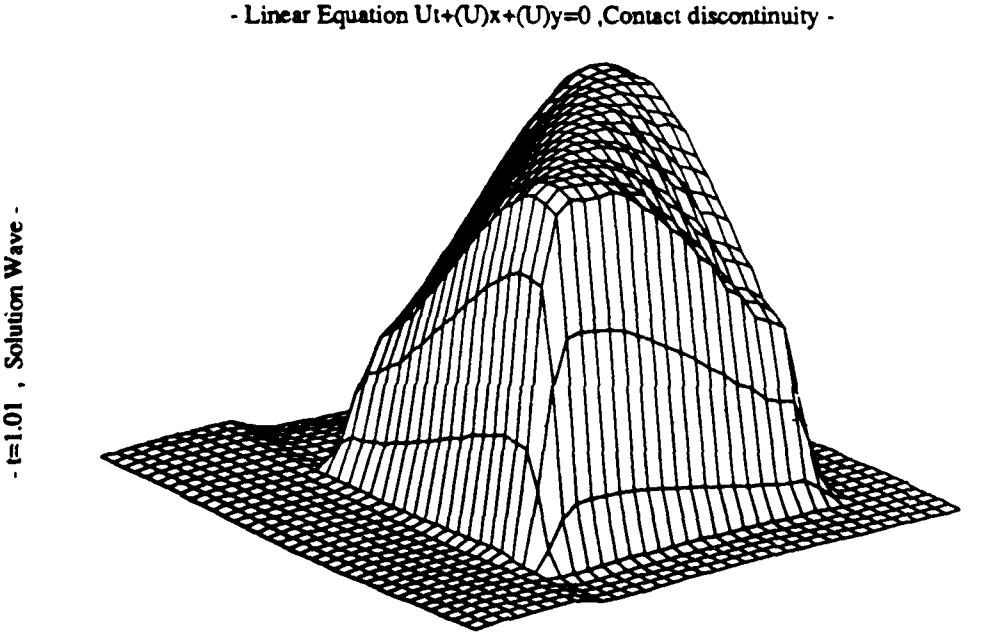
- Linear Equation Ut+(U)x+(U)y=0 ,Contact discontinuity -



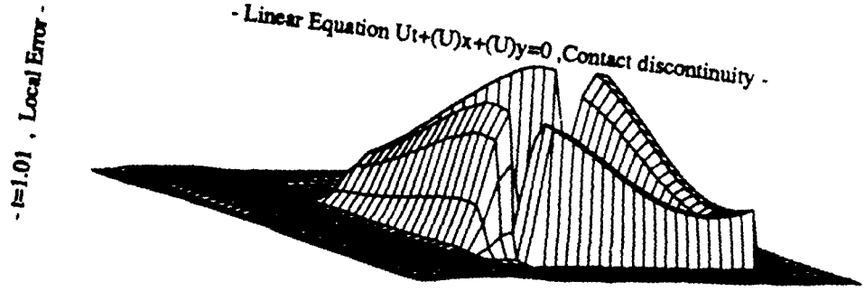
- (3,4) FM scheme , figure 5.3.1 -

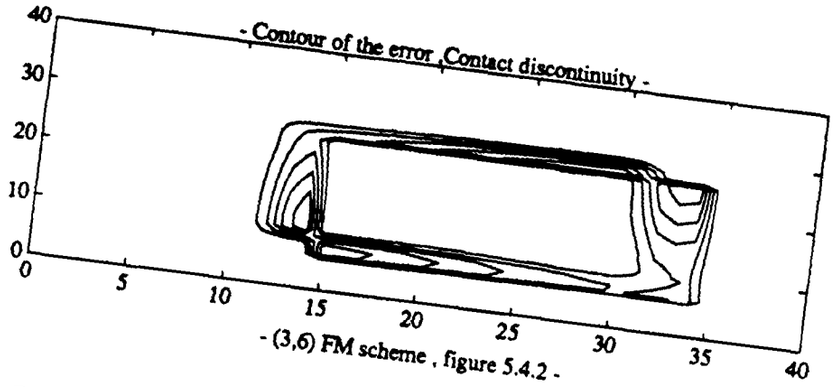


- (3,4) FM scheme , figure 5.3.2 -

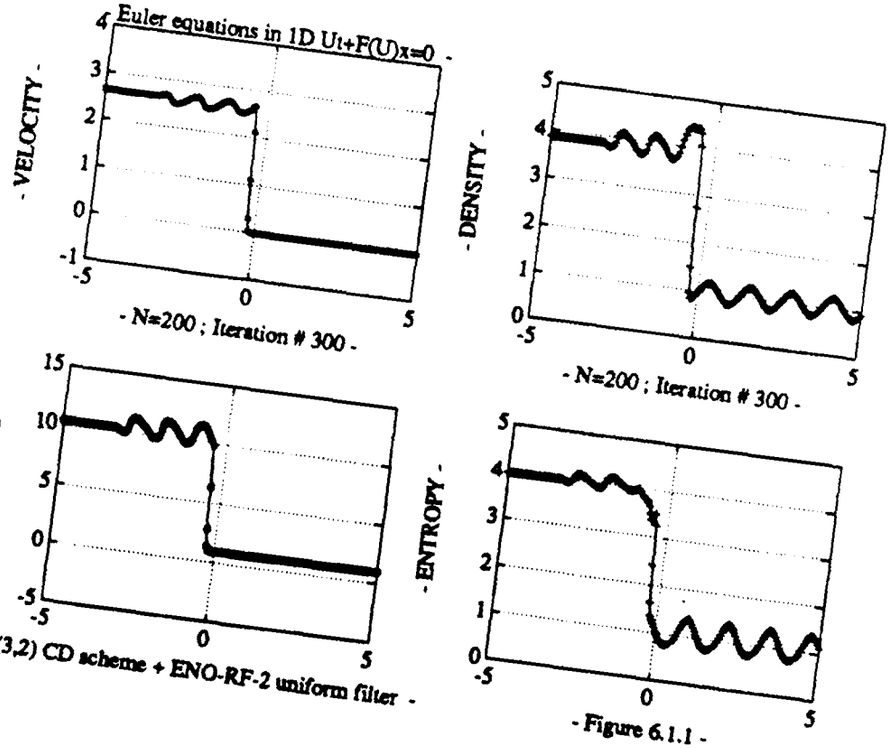


- (3,6) FM scheme , figure 5.4.1 -

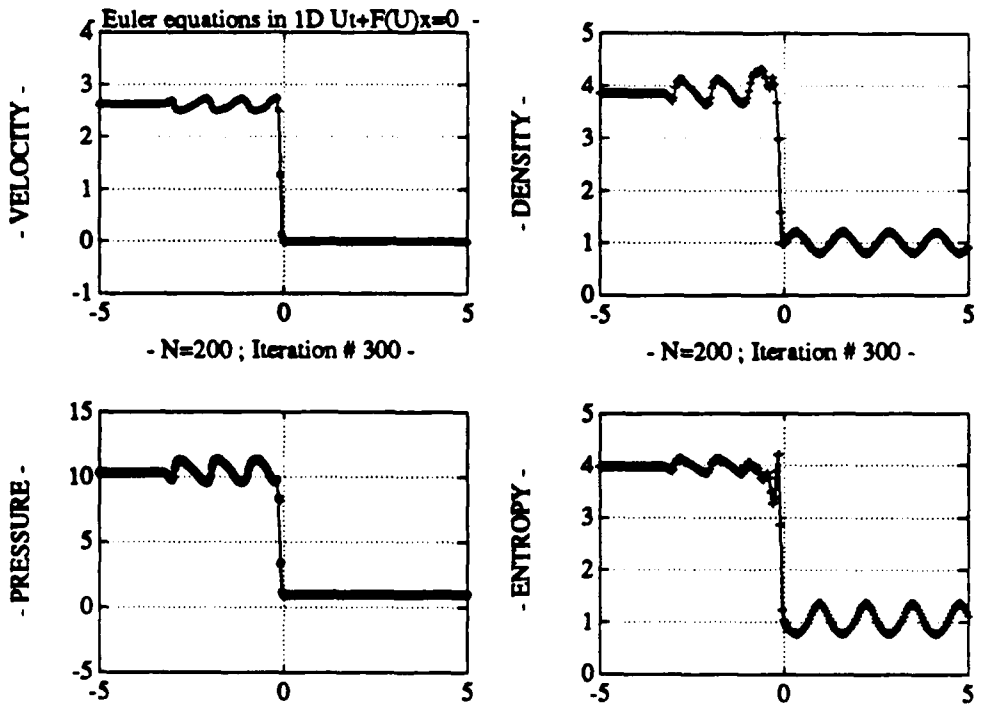




- (3,6) FM scheme , figure 5.4.2 -

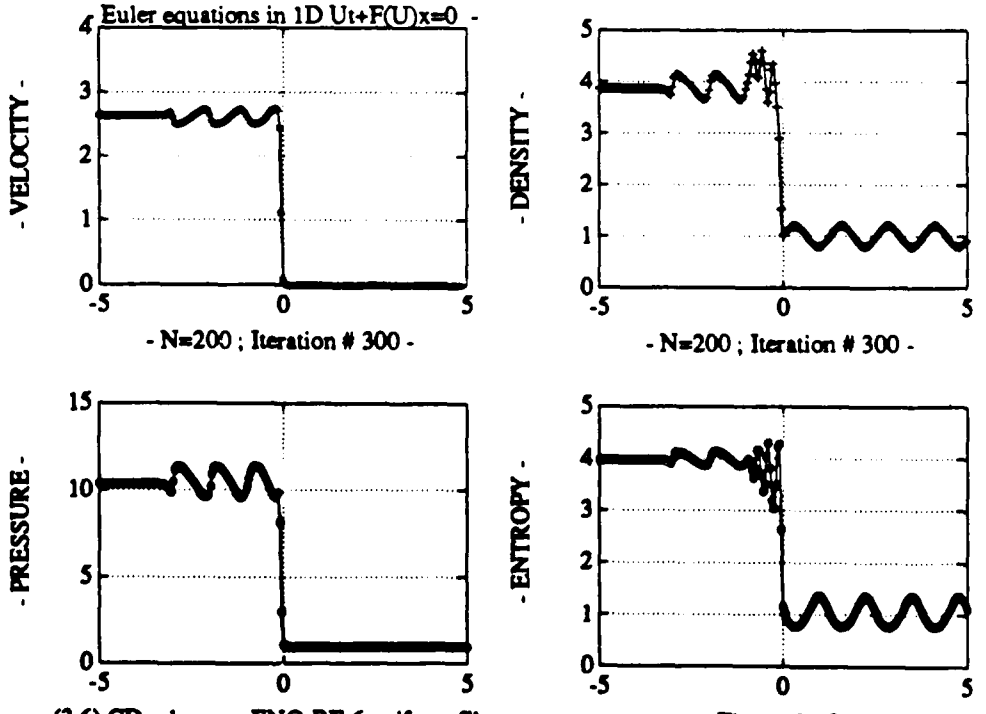


- Figure 6.1.1 -



- (3,4) CD scheme + ENO-RF-4 uniform filter -

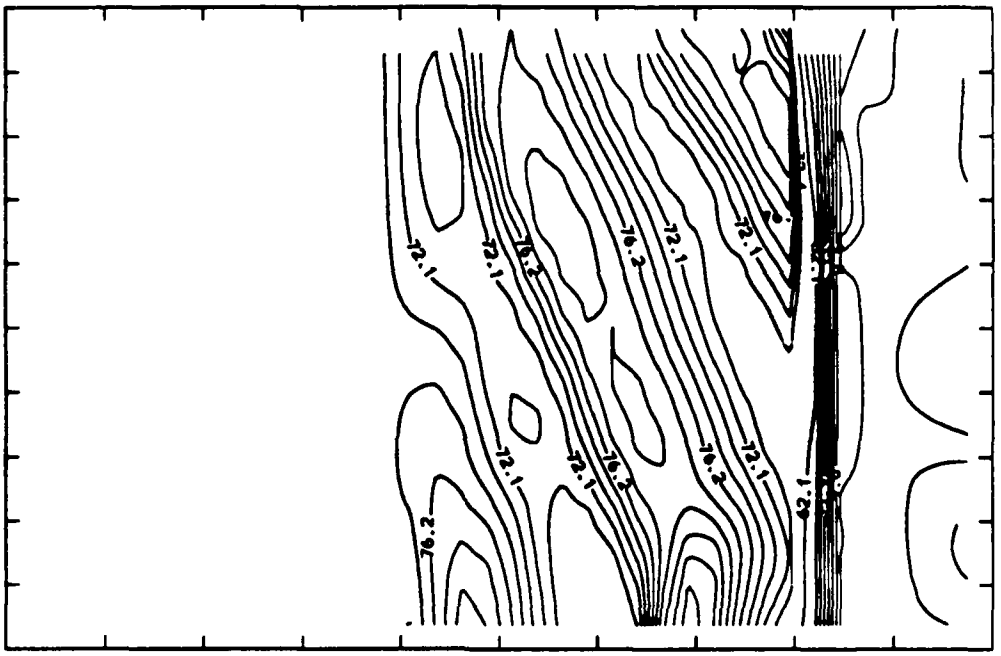
- Figure 6.1.2 -



- (3,6) CD scheme + ENO-RF-6 uniform filter -

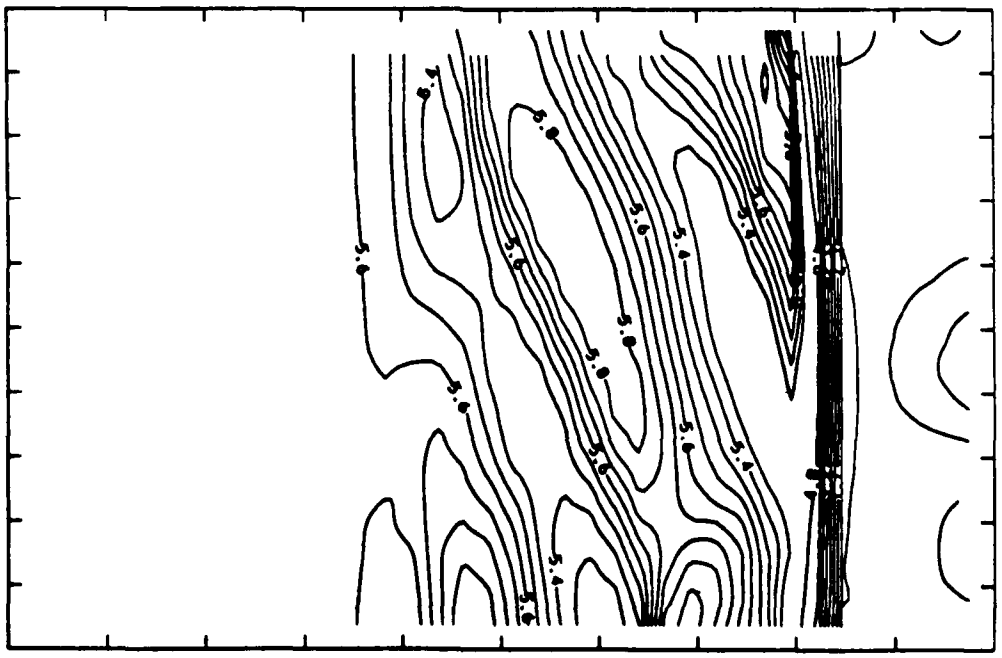
- Figure 6.1.3 -

PRESSURE - ITERATION NUMBER 400



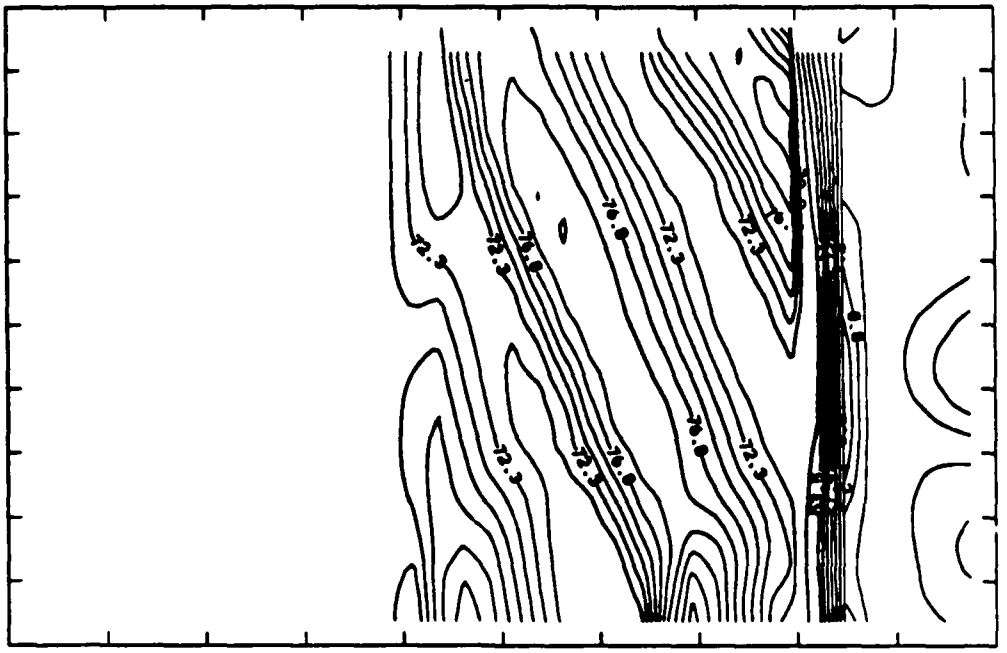
CONTOUR FROM 0.0000 TO 84.316 CONTOUR INTERVAL OF IRREGULAR
X INTERVAL= 0.29752 Y INTERVAL= 0.19753

DENSITY - ITERATION NUMBER 400



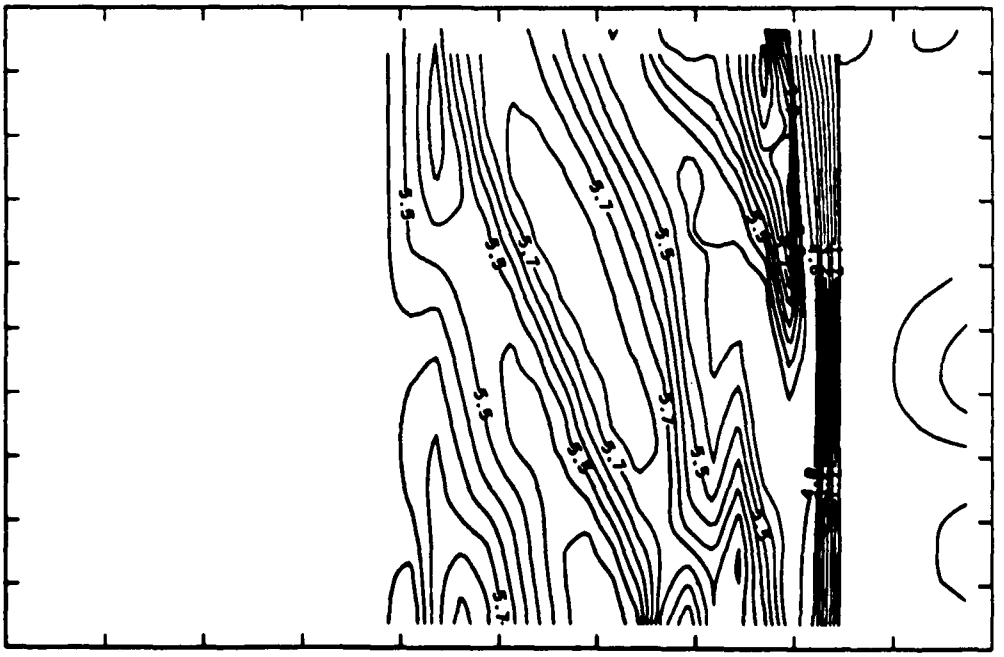
CONTOUR FROM 0.0000 TO 6.0071 CONTOUR INTERVAL OF IRREGULAR
X INTERVAL= 0.29752 Y INTERVAL= 0.19753

PRESSURE - ITERATION NUMBER 400



CONTOUR FROM 0.0000 TO 95.985 CONTOUR INTERVAL OF IRREGULAR
X INTERVAL= 0.29752 Y INTERVAL= 0.19753

DENSITY - ITERATION NUMBER 400



CONTOUR FROM 0.0000 TO 6.0310 CONTOUR INTERVAL OF IRREGULAR
X INTERVAL= 0.29752 Y INTERVAL= 0.19753